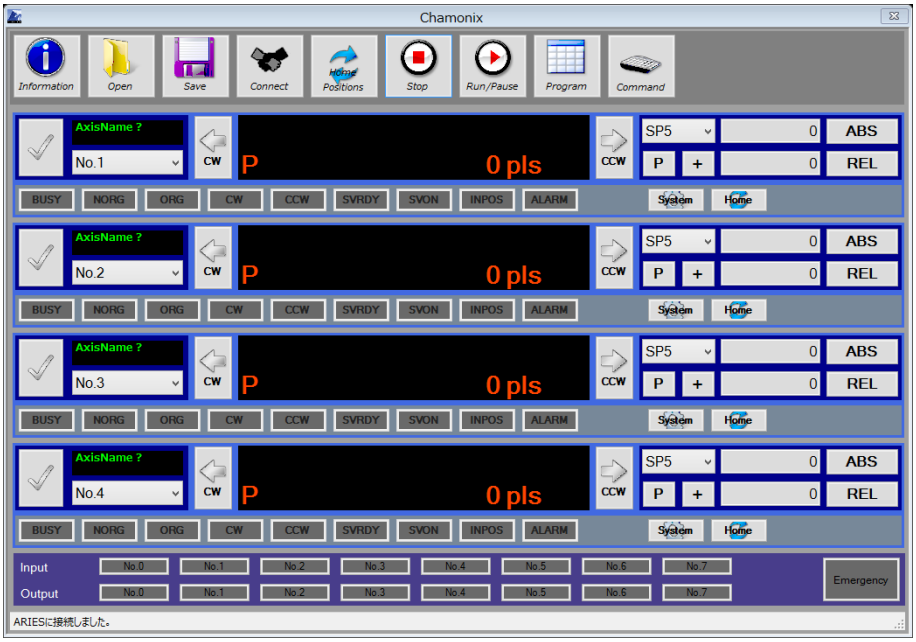
Chamonix

ユーザーズガイド

Rev. 1.0.2



神津精機株式会社

目次

目次 3

1. はじめに 8

1.1. Chamonix について 8

1.2. このユーザーズガイドについて 8

1.3. ご使用上の注意 9

1.4. 使用許諾契約の内容 10

1.5. 対応 OS 11

2. インストール 12

2.1. インストール CD 12

2.2. 「Setup」の起動 13

2.3. 旧バージョンの削除 14

2.4. 使用許諾契約の確認と同意 15

2.5. ユーザー情報の入力 16

2.6. インストールの実行 17

2.7. インストールの完了 18

2.8. インストールの確認 19

3. アンインストール 20

3.1. コントロールパネル 20

3.2. プログラムのアンインストール 21

3.3. 確認のダイアログ 22

4. コントローラとの配線 23

4.1. RS-232C 23

4.2. CRUX の通信設定 24
CRUX の通信設定用ロータリースイッチの設定 24
CRUX の通信設定用ロータリースイッチの位置 25
4.3. CRUX の結線方法 26
4.4. ARIES の通信設定 27
ARIES の通信設定用ロータリースイッチの位置 27
ARIES の通信設定用ロータリースイッチの設定 28
4.5. ARIES の結線方法 29
5. 起動してみましょう！ 30
5.1. デスクトップのアイコンから起動する 30
5.2. モダン UI から起動する 31
5.3. 保存したファイルから起動する 32
5.4. 起動画面 33
6. RS-232C でコントローラと接続する 34
6.1. COM ポートの選択 34
6.2. 接続確認 35
6.3. COM ポートの番号の確認方法 36
7. バージョン情報を確認しましょう 37
7.1. このアプリのバージョン情報を確認する 37
8. 原点復帰をおこないましょう 39
8.1. 全軸原点復帰 39
8.2. 原点復帰中のステータス 40
8.3. 単軸の原点復帰 41

9. ステージを動かしてみましょう 42
9.1. 制御する軸の選択 42
9.2. 軸の名前の入力 43
9.3. 現在位置表示の切り替え 44
9.4. 速度テーブルの選択 45
9.5. マニュアルでの移動 46
10. 位置決め 47
10.1. 絶対位置管理による位置決めの設定 47
10.2. 絶対位置管理による位置決めの実行 48
10.3. 相対位置管理による位置決めの設定 49
10.4. 相対位置管理による位置決めの実行 50
10.5. 多軸駆動を実行 51
11. 非常停止を受信 52
11.1. 非常停止信号を受信（ARIES) 52
11.2. 非常停止からの復帰（ARIES） 53
11.3. CRUX の場合 54
12. システムパラメータを変更する 55
12.1. パラメータ設定を起動する 55
12.2. パラメータを設定する 57
12.3. システムパラメータのファイルへの保存 59
12.4. パラメータの印刷 60
13. プログラマブルグリッド 61
13.1. プログラマブルグリッドの起動 61

13.2. プログラマブルグリッドの基本機能 62
13.3. プログラムを実行 63
13.4. プログラムの停止 64
13.5. プログラムデータの入出力 65
13.6. プログラマブルグリッドのコマンド 66
13.6.1. ORG: 原点復帰 66
13.6.2. ABS： 絶対位置管理による位置決め 67
13.6.3. REL: 相対位置管理による位置決め 68
13.6.4. WOT： 汎用出力の制御 69
13.6.5. FOR： 繰り返しループ 70
13.6.6. IF： 条件付きの実行 71
13.6.7. GOTO： 行のジャンプ 71
13.6.8. SBY： スタンバイ 72
13.6.9. プログラマブルグリッドのコマンド一覧 73
14. 直接コマンドを入力する 74
14.1. コマンドコントロールの起動 74
14.2. コマンドコントロールからコマンドを送信 75
15. API を利用して外部から制御しましょう 76
15.1. WindowsAPI による外部制御 76
15.2. 外部のアプリケーションからの制御 77
15.3. 外部のアプリから APS コマンドを送る 78
15.4. 外部のアプリから RPS コマンドを送る 80
15.5. 外部のアプリから STP コマンドを送る 82

15.6. 外部のアプリから ORG コマンドを送る 84
15.7. 外部のアプリからプログラマブルグリッドのスタンバイを解除する 86
15.8. 外部制御のサンプルプログラム 88
16. Chamonix を終了する 89
16.1. Chamonix を終了する 89
17. その他 90
17.1. 通信ログの保存場所 90
17.2. 通信ログを確認 91
17.3. 詳細なヘルプファイル 92
索引 94
改訂履歴 97

1. はじめに

1.1. Chamonix について

Chamonix（シャモニー）は神津精機製モーターコントローラを制御するアプリケーションです。神津精機製のモーターコントローラ、KOSMOS シリーズの ARIES 及び LYNX と旧製品 SC シリーズ[1]の SC-20、SC-21、SC-200、SC-400、SC-210、SC-410 を制御できます。

Chamonix は 4 軸まで同時に制御できる GUI と簡易的なマクロのようなプログラミンググリッドを備えています。また WindowsAPI を通して外部のアプリケーションと連携をとることも可能です。

コントローラとの通信には、RS-232C のシリアル通信を利用しています。

このアプリケーションは予告なく変更される場合があります。 ご利用にあたっては使用許諾条件をよくお読みください。

1.2. このユーザーズガイドについて

このガイドはモーターコントローラ制御アプリケーション「Chamonix（シャモニー）」を説明したものです。

「Chamonix」の性能ならびに機能を十分にご利用いただくために、よくお読みになり、十分理解した上でご使用ください。

また、いつでもご利用いただけますよう、は大切に保管してください。

本書は最終的にご使用いただく方のお手元に届くようご配慮ください

[1] SC シリーズをご利用の場合、ファームウェアのバージョンアップが必要な場合があります。

1.3. ご使用上の注意

始業時または操作時には、このアプリケーションが正常に動作していることを確認してください。

このアプリケーションが万一故障した場合の各種損害を防止するために十分な安全対策を施してご使用ください。

取扱説明書や別途取り交わした仕様書などに記載された以外の不適当な条件下でのご使用に起因する場合、機能及び性能の保証はできかねますのでご注意ください。

本製品を他の機器と組み合わせて使用する場合、使用条件や環境などにより、機能および性能が満足できない場合がありますので十分検討の上ご使用ください。

人体の保護を目的とした使用はしないでください。

他のアプリケーションと併用しないでください。機能および性能が満足できない場合があります。

1.4. 使用許諾契約の内容

このアプリケーションをご使用になる前に、以下の契約条項を十分にお読みください。

対象

•この使用許諾契約の対象は、パッケージとして提供される制御アプリケーション「Chamonix」とそれに付属する各種のライブラリ、およびそれらの使用方法等を記入したマニュアル等の付属するすべてのドキュメントとします。以下ではChamonixと付属するライブラリおよび付属するすべてのドキュメントを総称して「このアプリケーション」と記すものとします。ドキュメントには紙、電子ファイル等のすべてのメディアが含まれます。

使用権

•神津精機株式会社はお客様がこの契約条項のすべてに同意した場合に、ユーザーに対してこのアプリケーションの非独占的使用権のみを許諾します。この非独占的使用権では、神津精機株式会社から使用許諾を受けたユーザーは1台のコンピュータにこのアプリケーションをインストールし使用することができます。ユーザーは非独占的使用権以外の一切の権利を有しないものとします。この使用権には無償のユーザーサポートは含まれていないものとします。

著作権

•このアプリケーションと、このアプリケーションとライブラリに付属するすべてのドキュメントに関する著作権は、特に断りのない限り神津精機株式会社あるいは該当部分の著作権を保持する個人または法人に帰属します。これらは著作権に関する法律および国際条約によって保護されています。

禁止条項

•ユーザーは、このアプリケーションとライブラリあるいは付属するすべてのドキュメントを複製し、第三者への転売・配布をすることはできません。ユーザー自身のためにバックアップを保有する目的以外で、このアプリケーションとライブラリを複製することはできません。ユーザーがこのアプリケーションを改造、リバースエンジニアリング、逆アセンブルあるいは逆コンパイル等を行うことを固くお断りいたします。

免責

•神津精機株式会社は、このアプリケーションを運用した結果により生じた、ユーザーもしくは第三者の直接的、間接的あるいは波及効果によるあらゆる損害に対して、いかなる責任も負わないものとします。また、ユーザーもしくは第三者のデータ、プログラムその他の無体財産に関する損害、使用利益および得べかりし利益の喪失等に対しても神津精機株式会社は一切の責任を負わないものとします。

その他

•神津精機株式会社は、いつでもこのアプリケーションを更新できるものとします。更新版の提供条件は更新時に神津精機が定めるものとします。更新された場合、神津精機は更新前のこのアプリケーションとの互換性を更新版で提供する義務を負わないものとします。神津精機株式会社は、いつでもこの使用許諾条件を変更できるものとし、変更がなされた場合には、従前の使用許諾条件は破棄され、最新の使用許諾条件が適用されるものとします。

契約の解除

•お客様が本許諾事項に違反したとき、あるいはこの使用許諾条項に同意されない場合は、直ちに契約を解除し、ソフトウェア及び複製物を神津精機株式会社に返却もしくは廃棄していただくものとします。

1.5. 対応 OS

Chamonix を動かすためには Microsoft.NET4.0 が必要です。また Chamonix は Windows7®あるいは Windows8®を正式にサポートしています。[2]

WindowsXP®はサポート対象外です。WindowsXP®にインストールされた場合、正常な動作の保証いたしかねますのでご了解ください。

WindowsXP®にインストールする場合、必ずシステムは C ドライブにインストールされている必要があります。他のドライブにシステムがある場合、深刻なトラブルが発生する恐れがあります。

x86 と x64 のいずれにも対応しています。X86 の場合には Program Files に、x64 の場合は Program Files(x86)にインストールされます。

Chamonix は、Microsoft .NET Framework を使用して構築されています。インストールには Microsoft .NET Framework Full 4.5 以上が必要です。

[2] .NET Framewrk、Windows®は米国 Microsoft Corporation の米国、およびその他の国における商標または登録商標です。

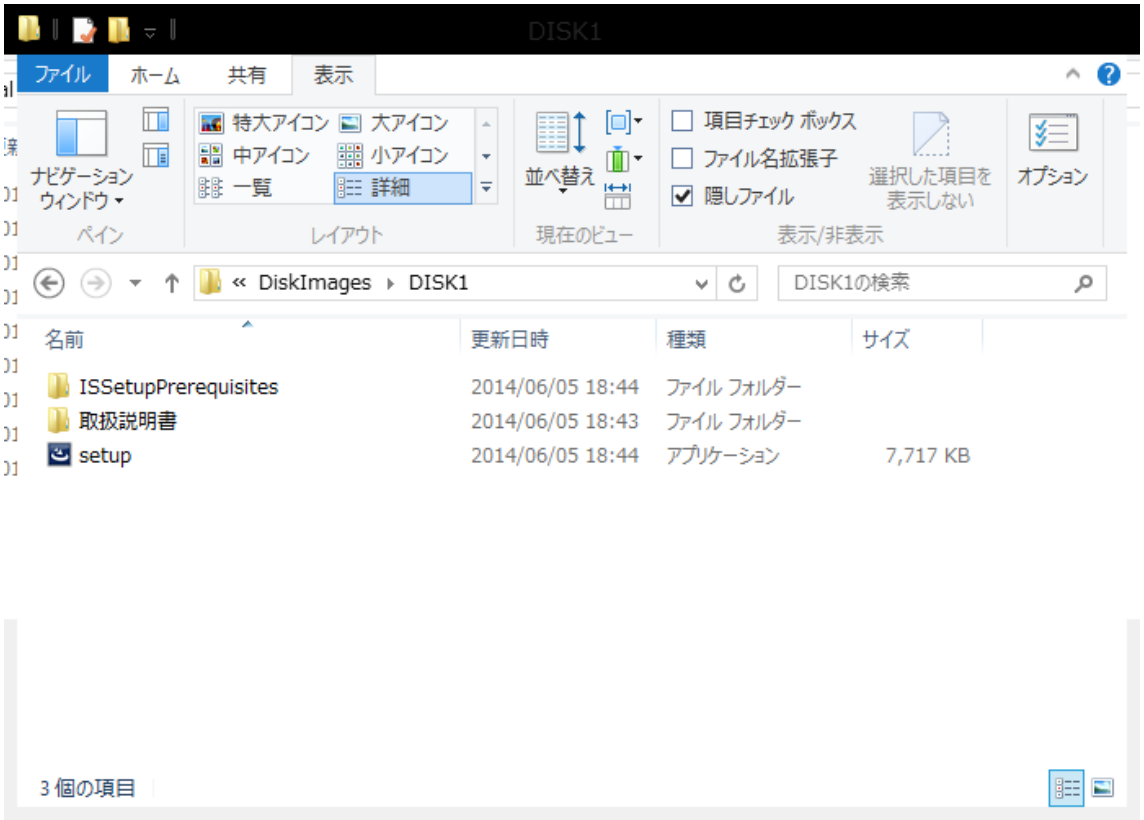
2. インストール

2.1. インストール CD

インストールをするためには管理者権限が必要です。インストール CD は以下のようなファイルで構成されています。インストールを実行するには「Setup」をクリックして実行してください。

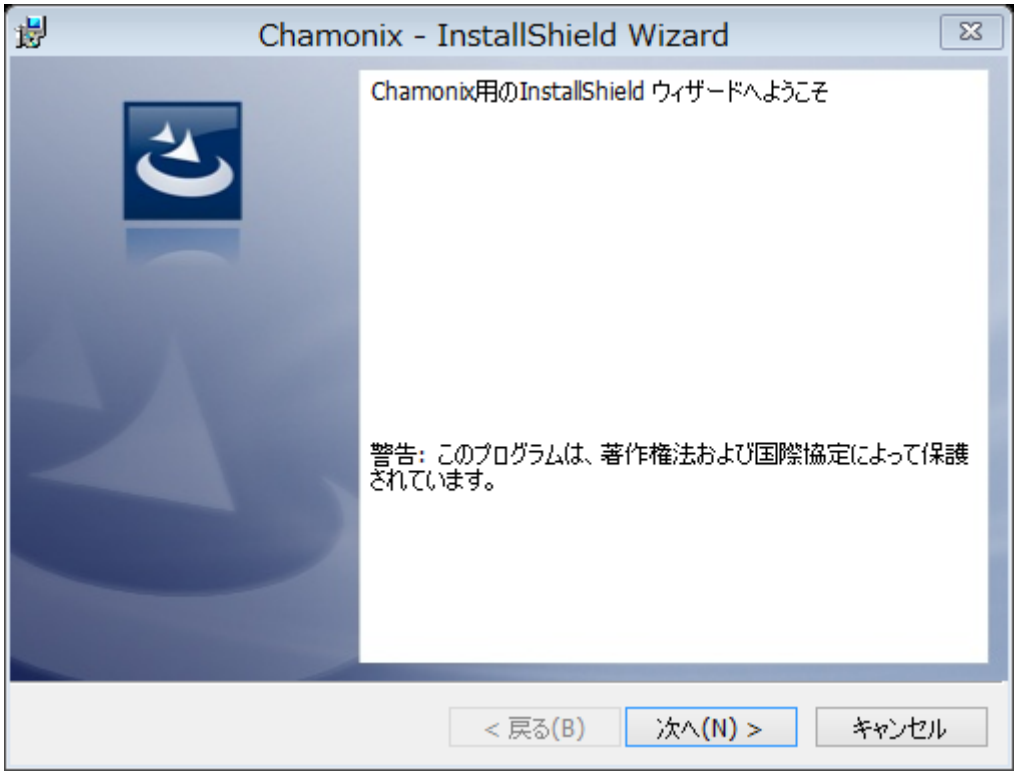
弊社 Web サイトからダウンロードいただいた場合、Zip ファイルで圧縮されています。解凍すると下記のようなフォルダが作成されます。

このフォルダのファイルは移動や変更をしないでください。



2.2. 「Setup」の起動

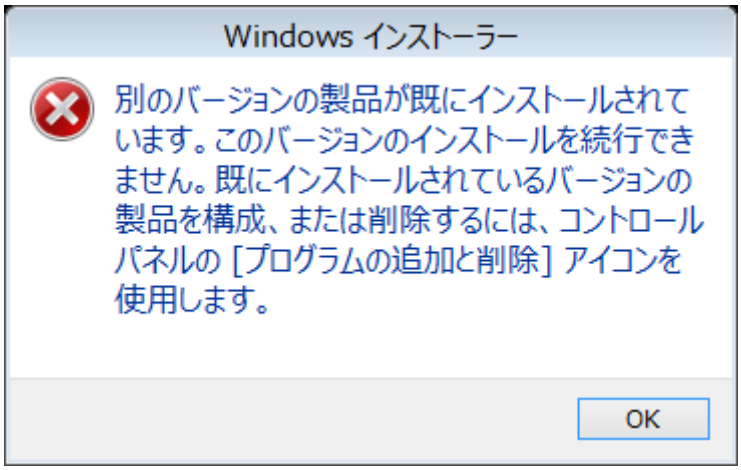
Setup.exe を起動すると Chamonix 用の InstallShield®[3]ウィザードが開きます。



このウィザードを利用してインストールを続けるのなら「次へ」を押してください。「キャンセル」を押すとインストールを終了します。つぎに使用許諾契約の確認と同意の画面が開きます。

[3] InstallShield は Flexera Software, Inc. および/ または InstallShield Co. Inc. の米国および/ または他の国における登録商標または商標です

2.3. 旧バージョンの削除



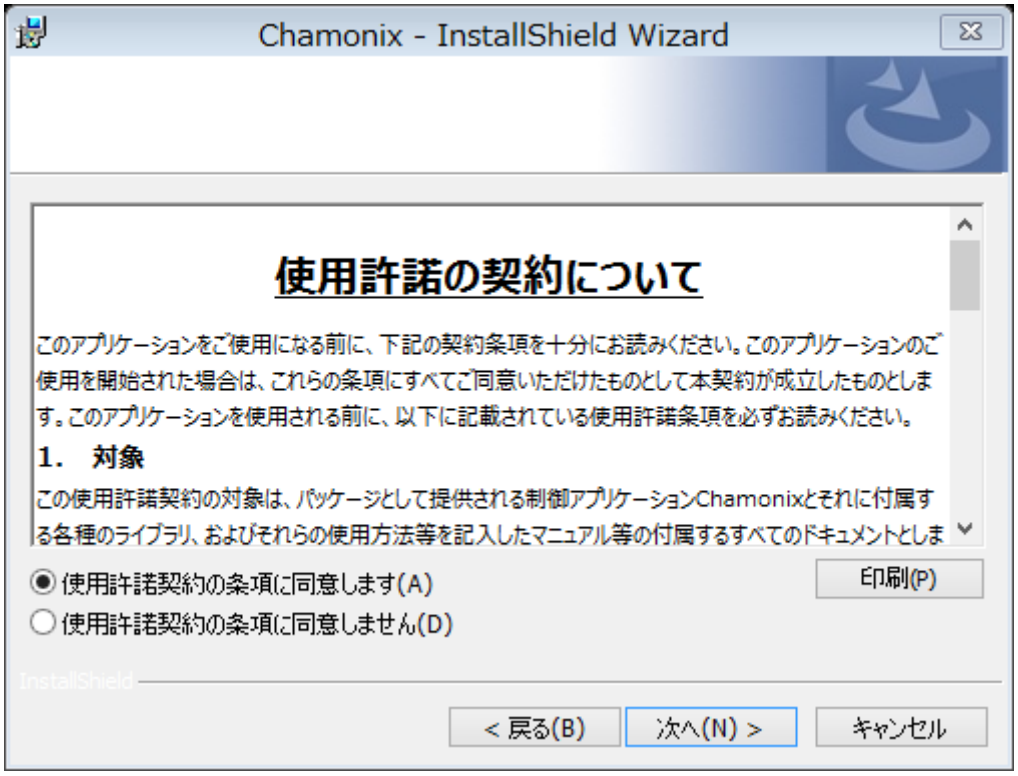
ChamonixあるいはChamonixを構成するダイナミックリンクライブラリの一部が既にインストールされている場合図のようなメッセージ[4]が表示される場合があります。

この場合には、既にインストールされている Chamonix をコントロールパネルの［プログラムの追加と削除］アイコンを使用して、削除する必要がりあります。

削除の方法はアンインストールで説明します。

[4] Microsoft Windows 8.1 の画面を示します。ご使用の OS によっては異なる場合があります。

2.4. 使用許諾契約の確認と同意



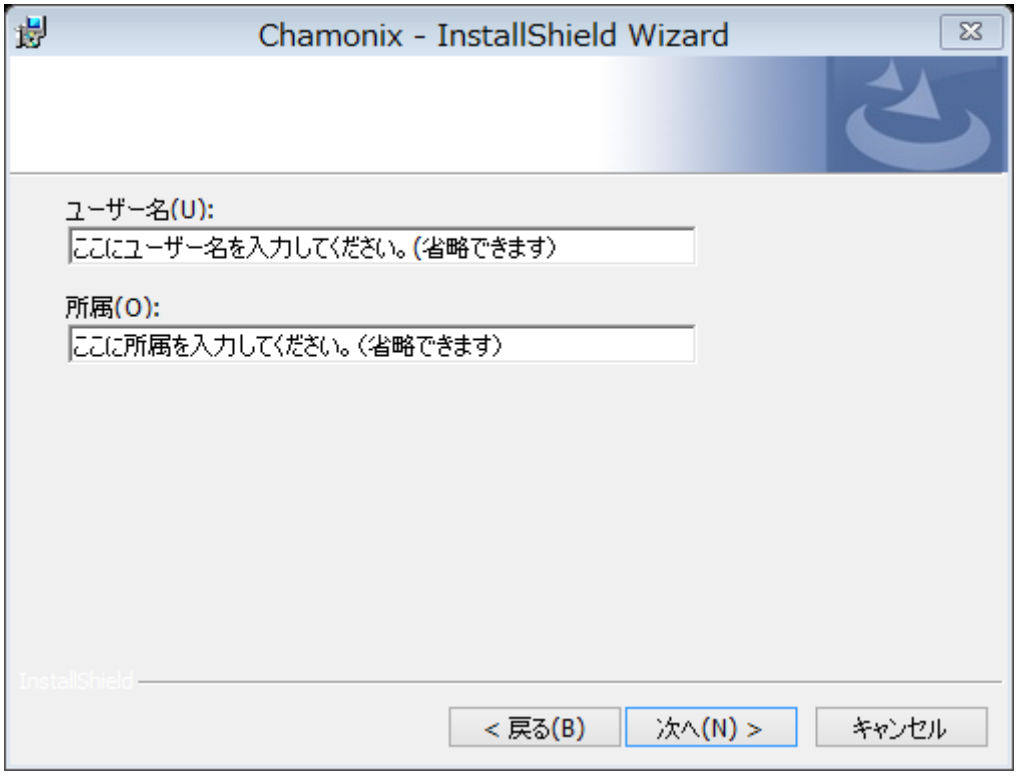
この使用許諾契約の内容に同意していただける場合は「使用許諾契約の条項に同意します」を選択してください。

同意していただけない場合は「使用許諾契約の条項に同意しません」を選択してください。この場合には、「次へ」は有効になりません。

「印刷」を押すと。[使用許諾の契約について」を印刷します。

使用許諾契約の条項に同意していただき、「次へ」を押すと「ユーザー情報の入力」へ移ります。

2.5. ユーザー情報の入力



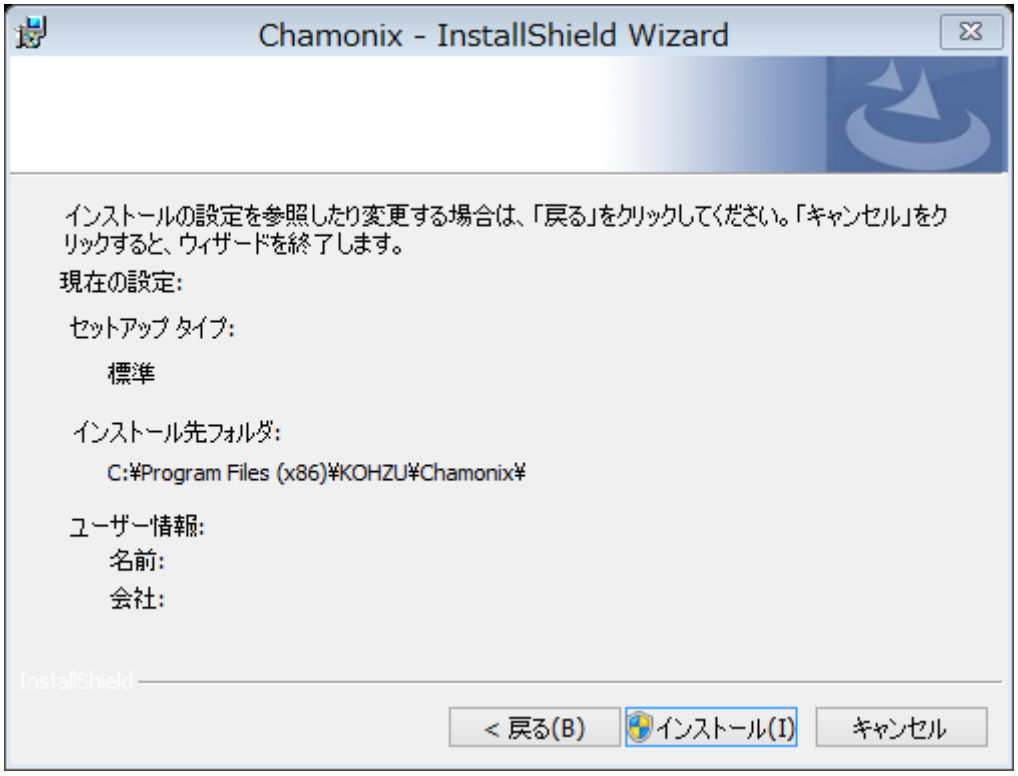
ユーザー名と所属を入力してください。

ユーザー情報の入力は任意です。

入力が終了したら「次へ」をクリックして下さい。もう一度使用許諾契約の内容を確認するためには「戻る」をクリックして下さい。

「キャンセル」をクリックするとインストールを中止します。

2.6. インストールの実行



インストールを実行するには「インストール」を押してください。

インストールを実行するためには、管理者権限が必要です。

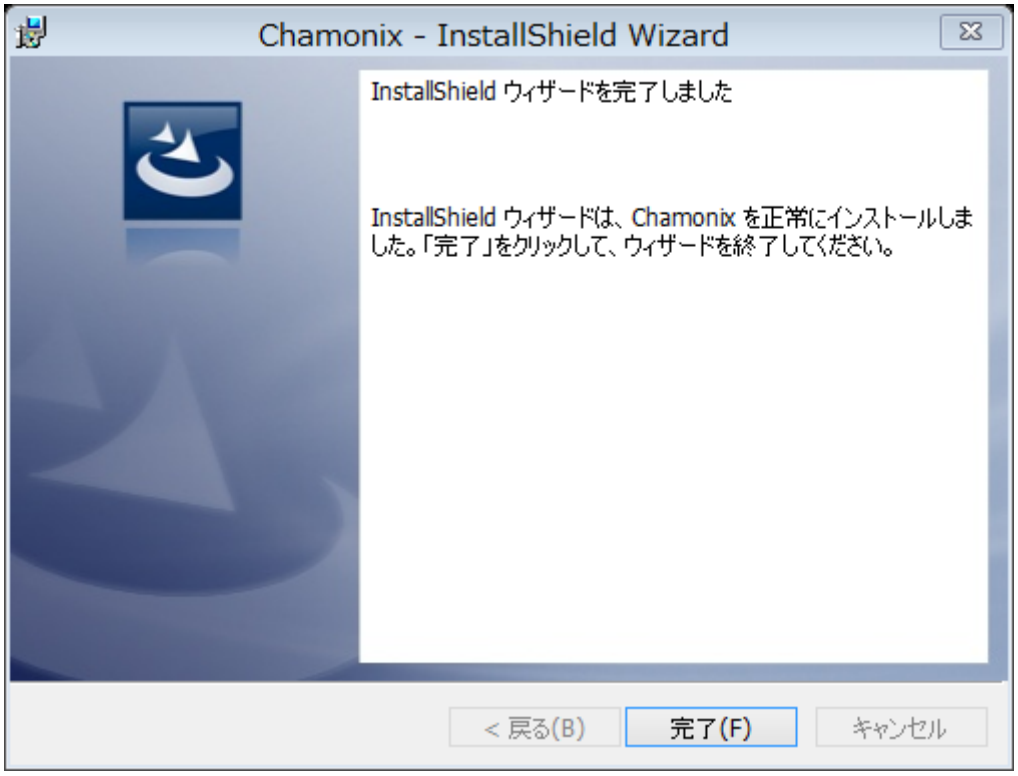
インストールを実行中に「ユーザーアカウントの制御」ダイアログが開き、インストールを実行するかどうかたずねられる場合があります。この場合には、Windows インストーラの当社のコードサイニング証明書をご確認いただき、「はい」を押してください。「いいえ」を押すとインストールは行われません。

インストール先ホルダーは変更できません。

ユーザー情報の入力へ戻るには。「戻る」を押してください。

「キャンセル」を押すとインストールウィザードを終了します。

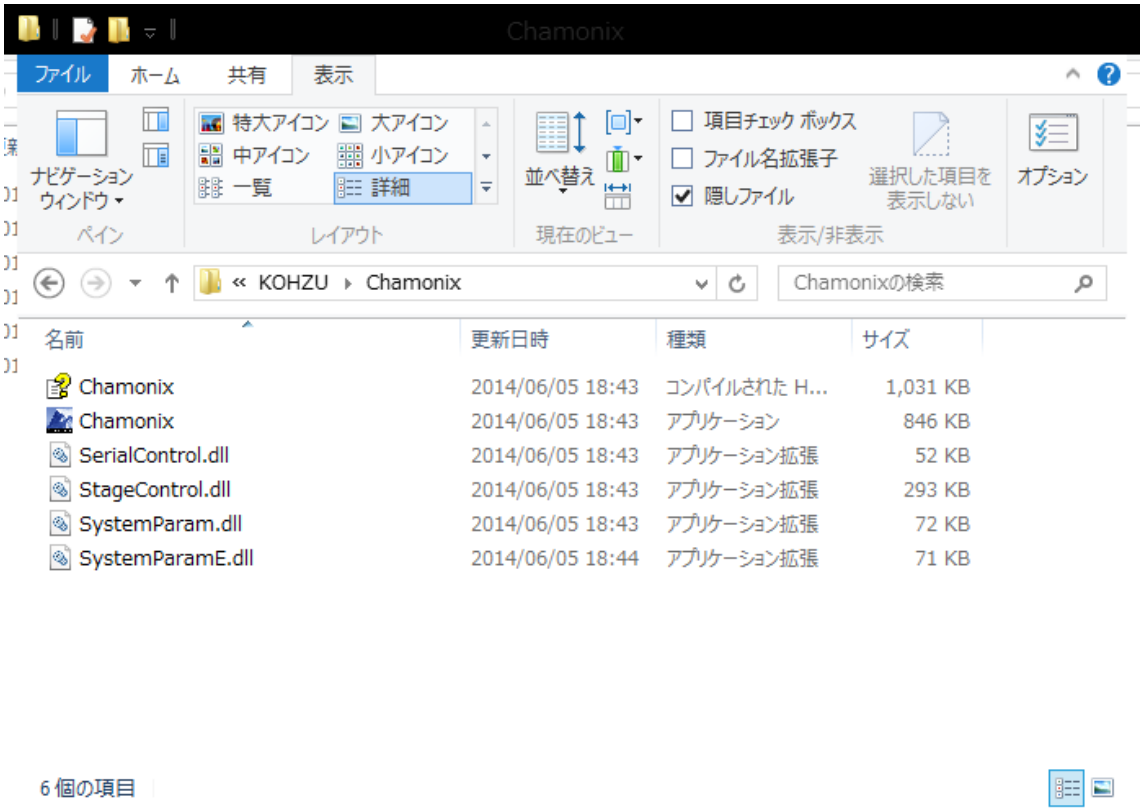
2.7. インストールの完了



前の画面で「インストール」を押すと、Chamonix のインストールが実行されます。上手の画面が表示をされたら正常にインストールされています。

「完了」をクリックするとウィザードは終了します。

2.8. インストールの確認



インストールが終了すると図のように C:¥Program Files(x86)¥KOHZU¥Chamonix フォルダに上手の 6 ファイルが生成されます[5]。

[5] Microsoft Windows 8.1 の画面を示します。ご使用の OS によっては異なる場合があります。

3. アンインストール

3.1. コントロールパネル

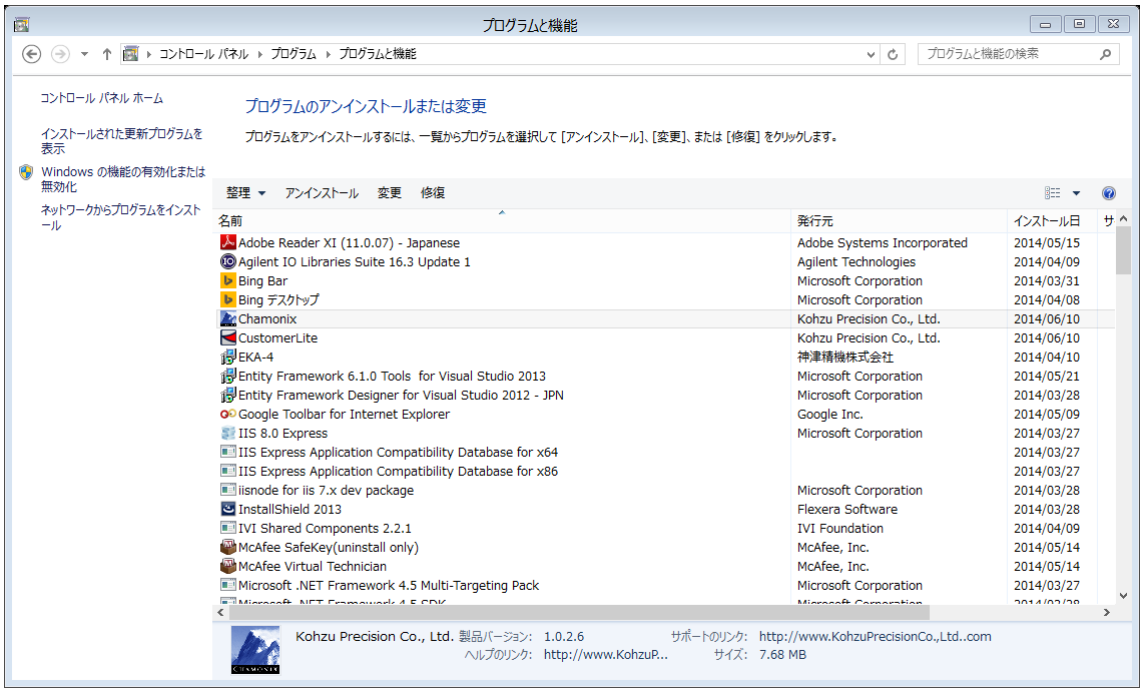
「コントロールパネル」[6]の「プログラム」を開きます。つぎに「プログラムの機能」をクリックします。

あるいは「コントロールパネル」から直接、「プログラムのアンインストール」をクリックします。

次ページのような「プログラムのアンインストールまたは変更」が開きます。

[6] 右図は Microsoft Windows 8.1 のコントロールパネルの例を示します。

3.2. プログラムのアンインストール

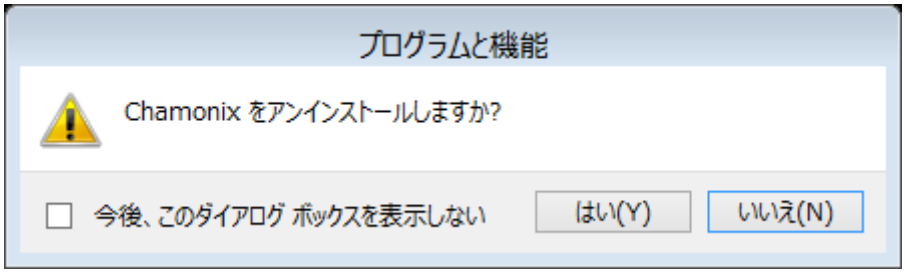


表示されたプログラムの一覧から Chamonix を選択します。

Chamonix をアンインストールする場合には、「アンインストール」をクリックします。
⇨確認のためのメッセージダイアログが表示されます。

Chamonix を修復する場合には、「修復」をクリックします。

3.3. 確認のダイアログ



メッセージダイアログの「はい」を押すとアンインストールされます。

アンインストールではインストールした Chamonix.exe、StageControl.dll、SerialControl.dll、SystemParam.dll や Chamonix.chm は消去されますが、インストール後にアプリケーションによって生成された SerialContlo_COM*.log や config.bin は消去されません。

これらのファイルを消去するには、エクスプローラなどから直接ファイルを消去する要があります。

該当するファイルは「C:¥Program Files(x86)[7]¥Common Files¥KOHZU」内に格納されています。

[7] 32 ビット版の OS の場合は C:¥Program Files ¥Common Files¥KOHZU¥Chamonix です。

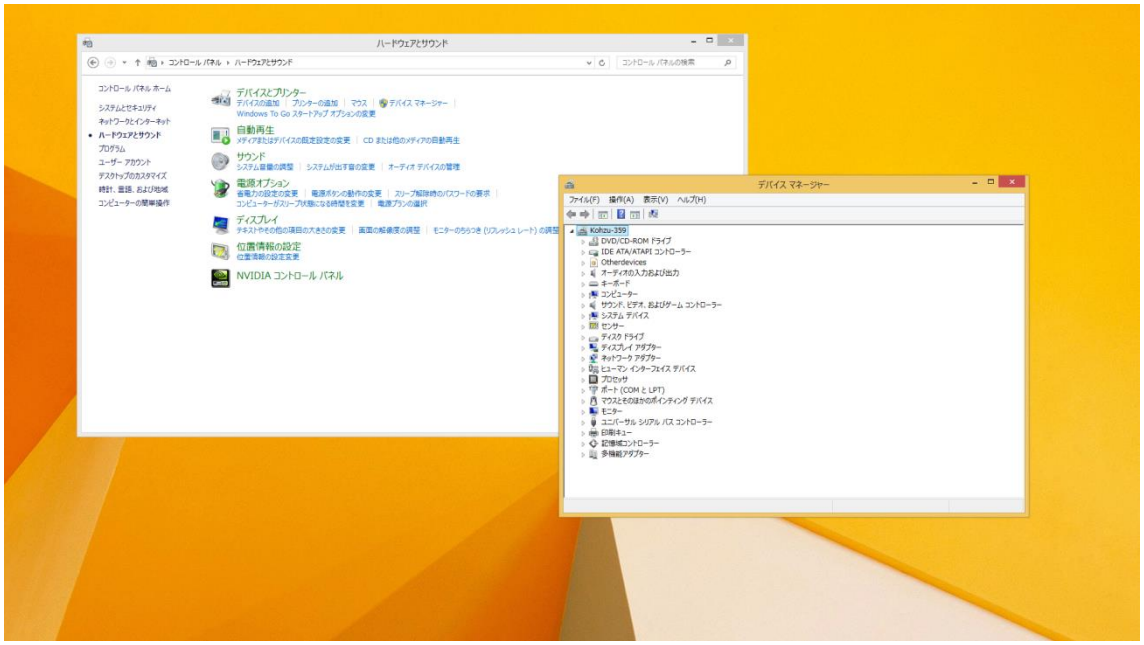
4. コントローラとの配線

4.1. RS-232C

Chamonix は RS-232C と USB（仮想 COM ポート）[8]のみの対応です。

あらかじめ Controller が接続されているポートをデバイスマネージャーで確認してください。

「コントロールパネル→ハードウェアとサウンド→デバイスマネージャー」で確認できます。デバイスマネージャーの起動には管理者権限が必要です。



[8] USB 仮想 COM ポートは CRUX のみで使用できます。

4.2. CRUX の通信設定

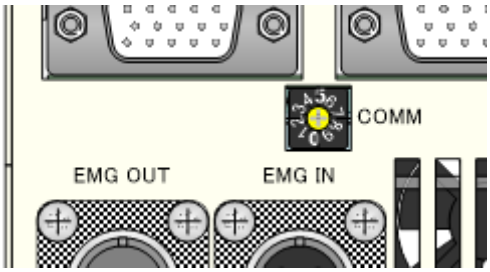
CRUX はリアパネルにあるロータリースイッチ（COMM）にて通信の条件を設定・変更することができます。
初期設定は 4（通常コマンド USB モード）です。

詳しくは CRUX/CRUX-A ユーザーズマニュアルをご覧ください。

※RS-232C 通信において速度(baud)以外の設定は以下のように設定してください。

パリティ	**NON**
語長	8bit
ストップビット	1

CRUX の通信設定用ロータリースイッチの設定



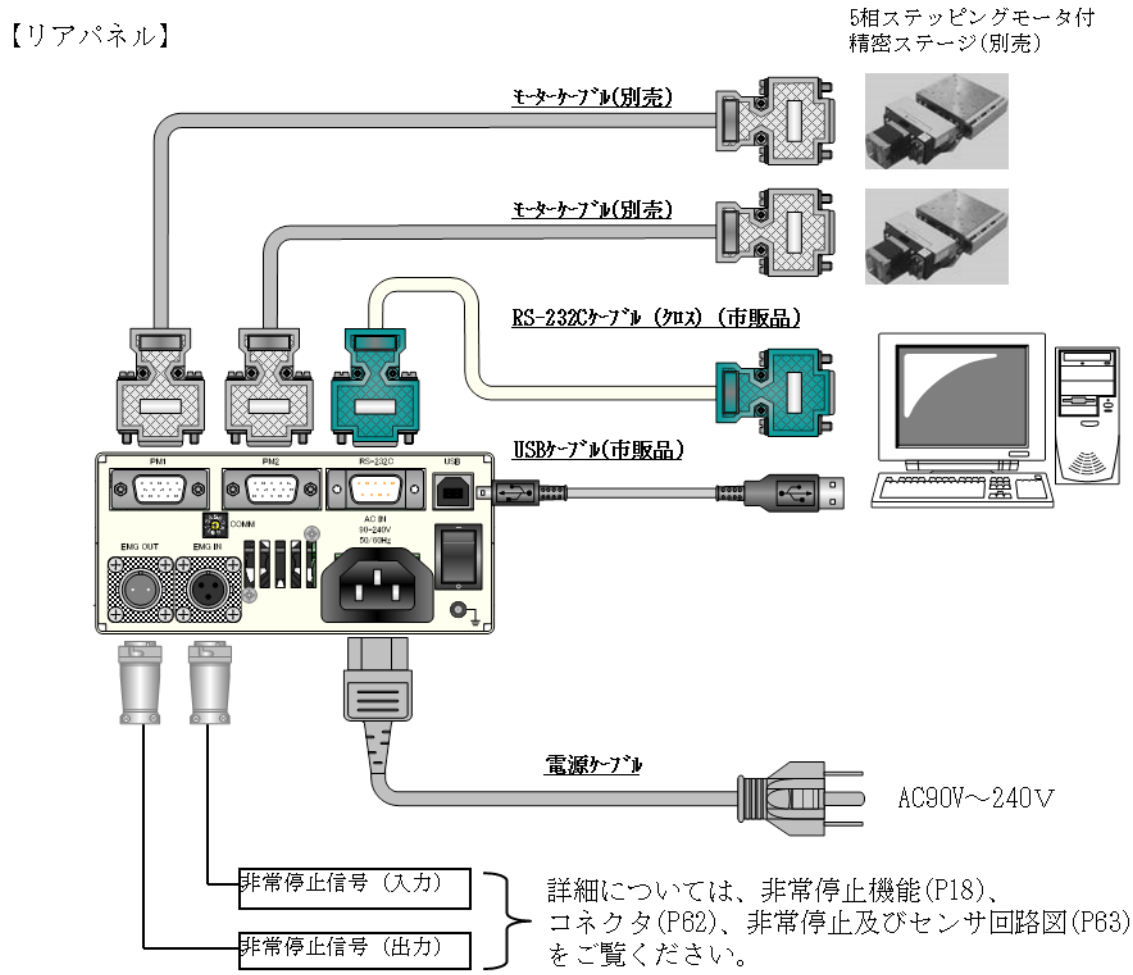
CRUX の通信設定用ロータリースイッチの位置

通信	通信設定		
ﾓｰﾄﾞ	RS-232C速度	USB	
0	38400	*	通常コマンドﾓｰﾄﾞ
1	57600	*	
2	19200	*	
3	9600	*	
4	*	USB	
5	38400	*	簡単コマンドﾓｰﾄﾞ
6	57600	*	
7	19200	*	
8	9600	*	
9	*	USB	

4.3. CRUX の結線方法

> 結線の抜き差しを行う際には、必ず本体の電源を切った状態で行ってください。

CRUX と、外部機器との接続/結線を説明します。



詳しくは CRUX/CRUX-A ユーザーズマニュアルをご覧ください。

4.4. ARIES の通信設定

通信設定を行う際には、必ず本体の電源を切った状態で行ってください。

ARIES はリアパネルにあるロータリースイッチ（COMM）にて通信の条件を設定・変更を行うことができます。初期設定はモード 4(RS-232C　115200baud)です。

詳しくは ARIES/LYNX ユーザーズマニュアルをご覧ください。

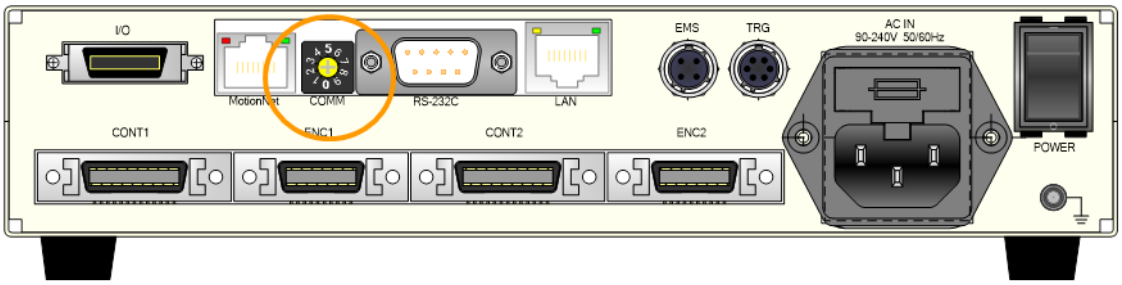
※RS-232C 通信において速度(baud)以外の設定は以下のように設定してください。

パリティ	**NON**
語長	8bit
ストップビット	1

ARIES の通信設定用ロータリースイッチの位置

通信設定用ロータリースイッチは以下の位置にあります。

【リアパネル】



ARIES の通信設定用ロータリースイッチの設定

設定は下表の通りです。（モード 6～9 使用不可）

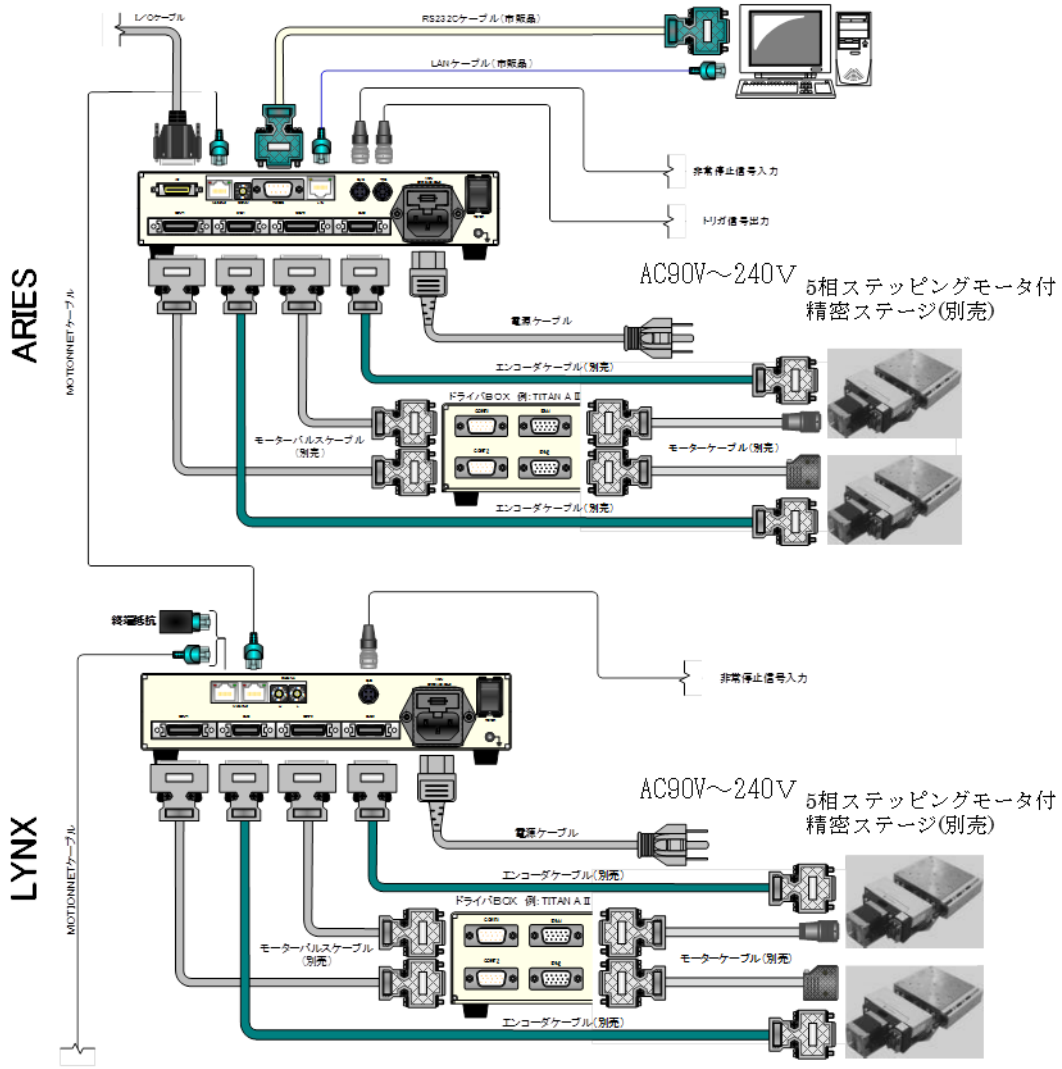
通信	通信設定	
ﾓｰﾄﾞ	RS-232C速度 (baud)	LAN
0	9600	*
1	19200	*
2	38400	*
3	57600	*
4	115200	*
5	*	LAN
6	*	*
7	*	*
8	*	*
9	*	*

4.5. ARIES の結線方法

結線の抜き差しを行う際には、必ず本体の電源を切った状態で行ってください。

ARIES と、外部機器との接続/結線を説明します。

【リアパネル】

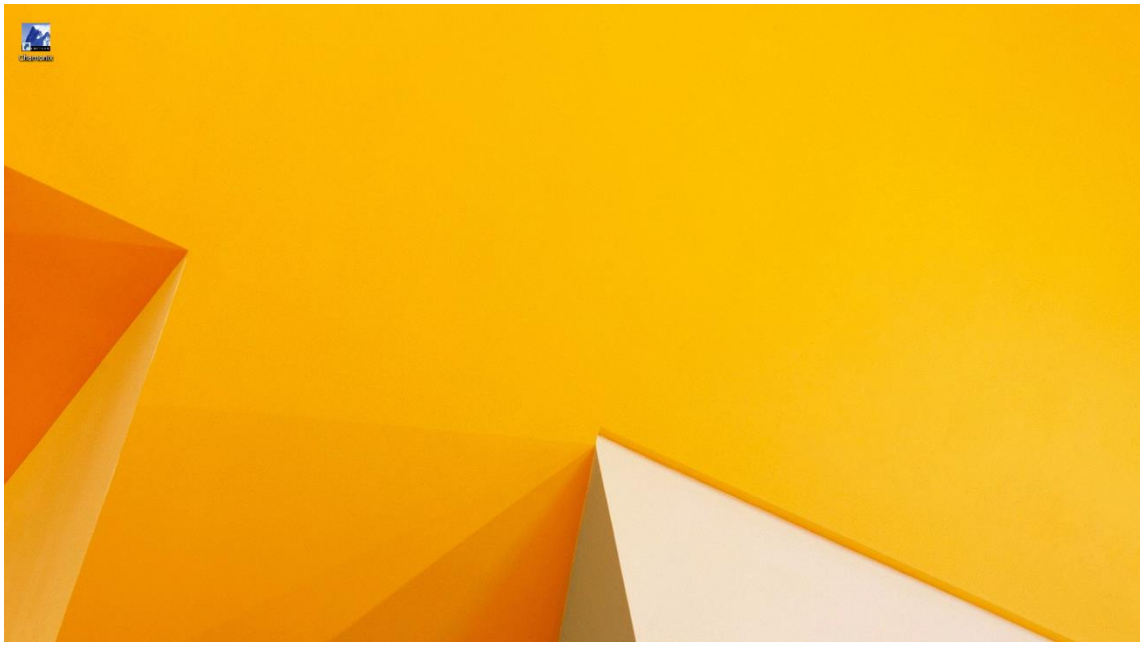


※Motionnet®ケーブル接続間にハブは使用しないでください。

詳しくは ARIES/LYNX ユーザーズマニュアルをご覧ください。

5. 起動してみましょう！

5.1. デスクトップのアイコンから起動する



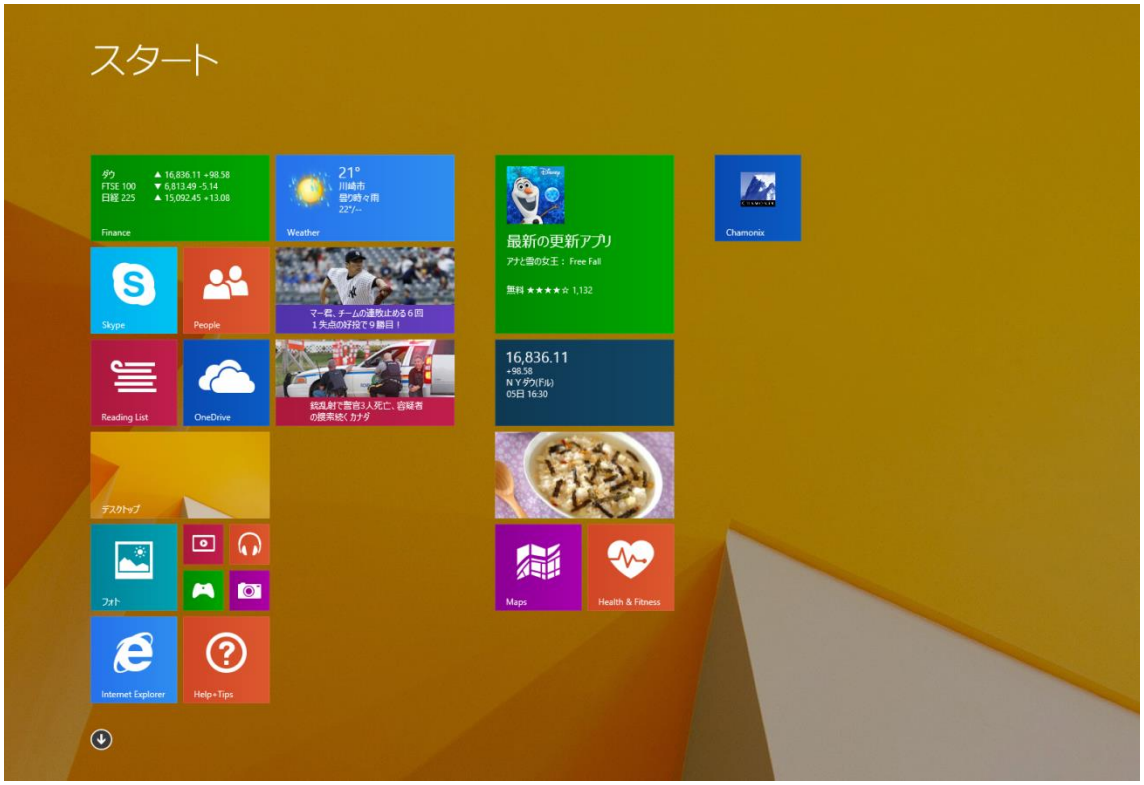
インストールが成功すると、デスクトップには Chamonix のアイコンが生成されます。

このアイコンをクリックすると前回終了時の時の通信設定で起動します。

初めて起動する場合は、デフォルトの通信設定で起動します。

デフォルトでは COM ポートの番号は「COM1」、ボーレートは「115200baud」です。

5.2. モダン UI から起動する

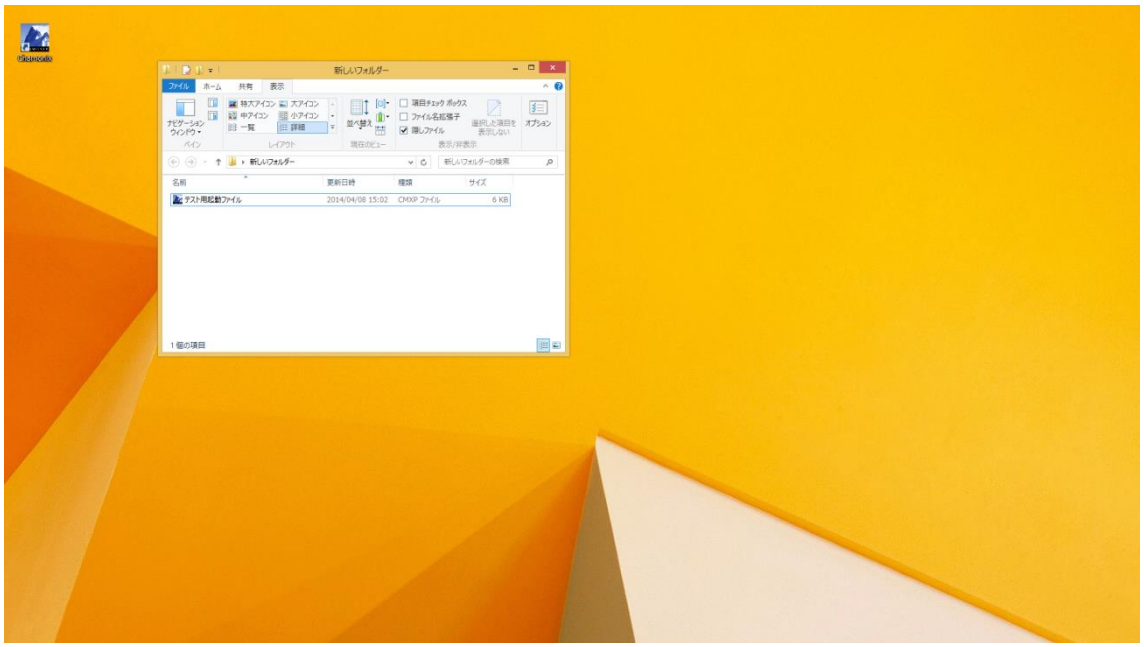


Window　8 以降の場合、モダン UI からも起動できます。

デスクトップからの起動と同様に、前回のパラメータで起動します。初めて起動する場合にはデフォルトの通信設定で起動します。

デフォルトでは COM ポートの番号は「COM1」、ボーレートは「115200baud」です。

5.3. 保存したファイルから起動する



保存した起動ファイルから起動できます。

起動ファイルにはポート番号、ボーレートなどの情報が保存されています。

保存した起動ファイルの拡張子は「.cmxp」です。

5.4. 起動画面

起動中、上記の画面が表示されます。

インストール時にはポート番号は COM1、ボーレードは 115200baud がデフォルトとして設定されています。

既に起動をしたことがある場合には、前回終了時の情報で起動します。

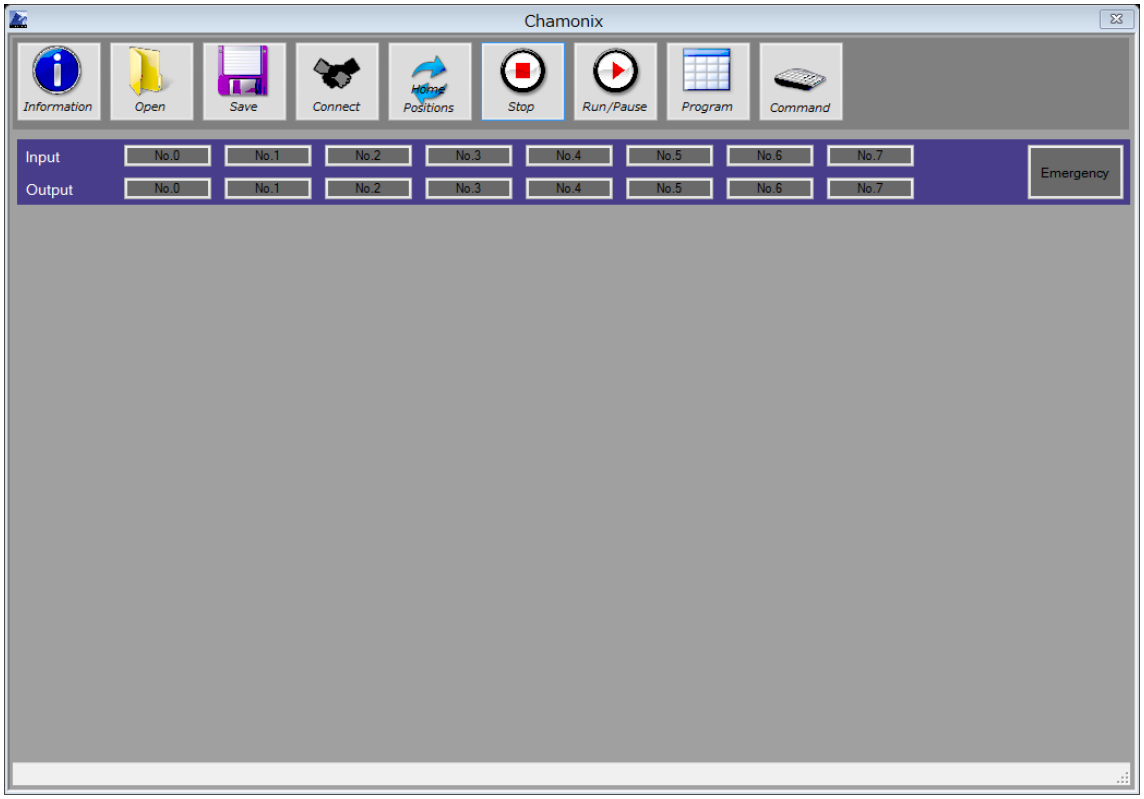
この値は「C:¥Program Files (x86)[9]¥Common Files¥KOHZU¥Chamonix」[10]内の Config ファイルに格納されています。

[9] 32 ビット版の OS の場合は C:¥Program Files ¥Common Files¥KOHZU¥Chamonix です。

[10] このファイルを編集すると Chamonix の起動が行えなくなる場合があります。

6. RS-232C でコントローラと接続する

6.1. COM ポートの選択



COM ポートが選択されていないと図のように駆動軸の制御画面が表示されません。

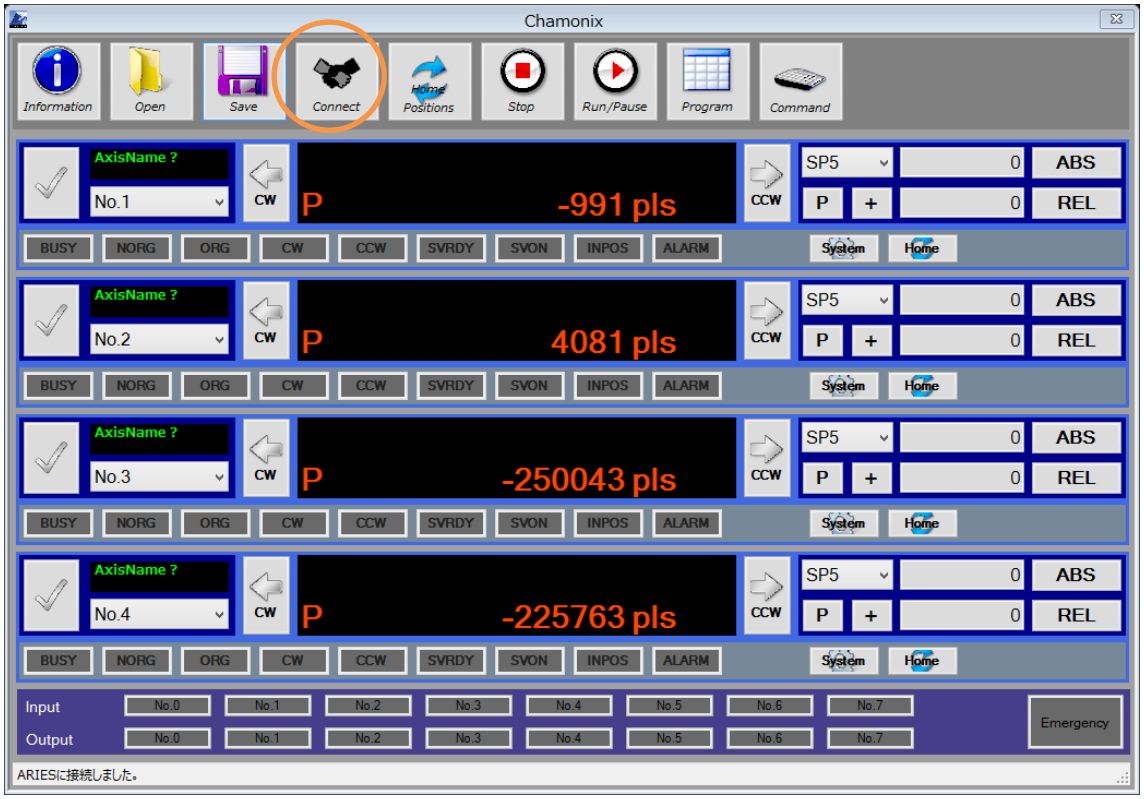
Connect ボタンを右クリックしてメニューを表示して COM ポートを設定して下さい。

弊社のコントローラの RS-232C 通信のボーレートは 115200baud で出荷されています。詳しくは各コントローラの取扱説明書をご覧ください。

Connect ボタンを左クリックすると接続確認と初期化を行います。

COM ポートが選択されて接続確認が正常に終了した後に、このアプリケーションを閉じて、新たに起動すると設定した値で起動します。

6.2. 接続確認



Connect ボタンを押すと接続の確認と初期化を行います。

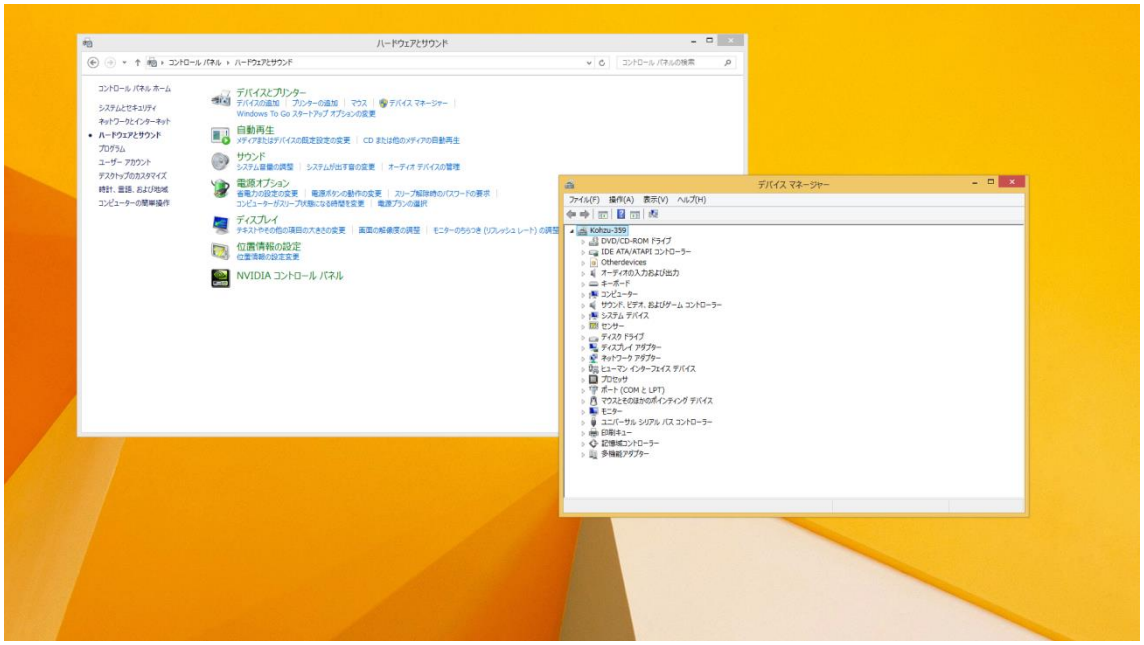
起動時に COM ポートの番号とボーレートの設定が正常に読み込まれた場合には自動で接続の確認と初期化が行われます。

接続の確認と初期化に時間がかかる場合には、5 秒以上起動に時間がかかる場合があります。

接続が確認された場合、図のように駆動軸の制御画面（StageControl）が表示されます。駆動軸制御画面は最大で 4 軸まで表示されます[11]。

[11] CRUX をご使用の場合、あるいは ARIES にスレーブコントローラ LYNX を接続していない場合など、総軸数が 2 軸の場合は 2 軸だけが表示されます。

6.3. COM ポートの番号の確認方法

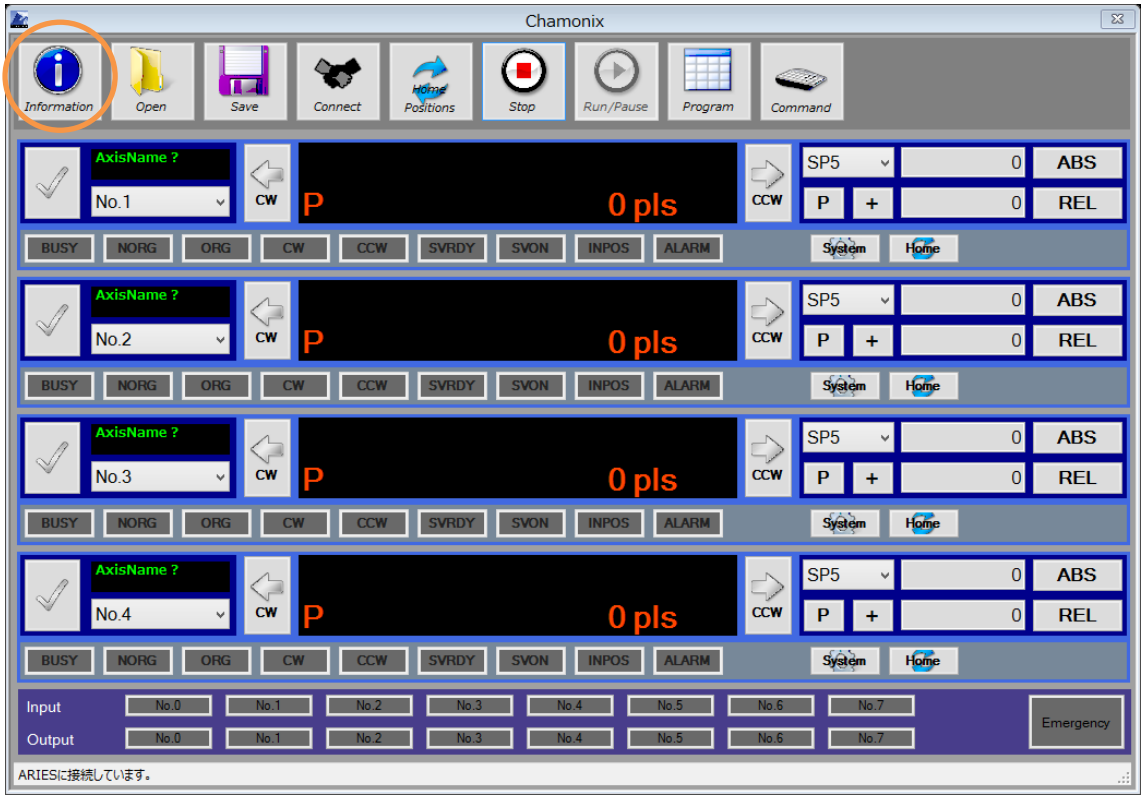


コントローラが接続されているポートをデバイスマネージャーで確認してください。

「コントロールパネル→ハードウェアとサウンド→デバイスマネージャー」で確認できます。デバイスマネージャーの起動には管理者権限が必要です。

7. バージョン情報を確認しましょう

7.1. このアプリのバージョン情報を確認する

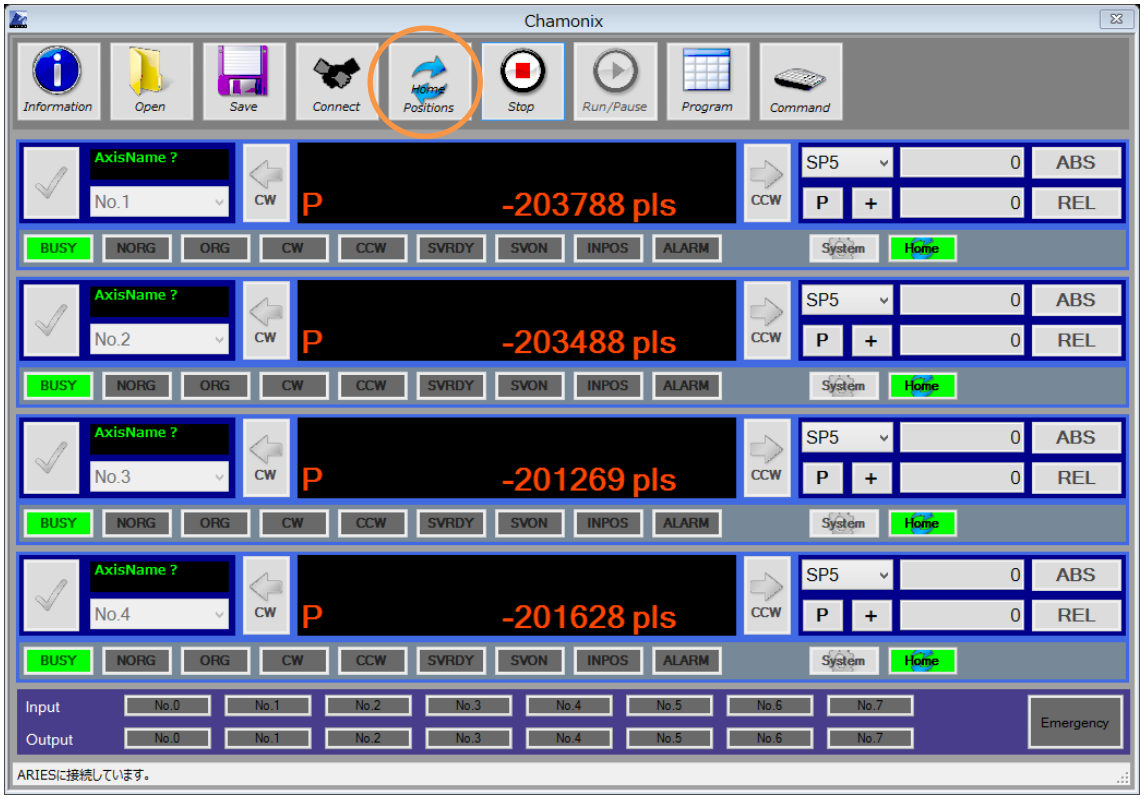


Information ボタンをクリックすると次頁のようにバージョン情報を確認できます。

Chamonix のバージョン情報
Chamonix
バージョン 1.0.2.6
Copyright © 2014 Kohzu Precision Co., Ltd.
神津精機株式会社（Kohzu Precision Co., Ltd.）
ChamonixはWindowsベースのアプリケーションです。
このアプリケーションは著作権に関する法律および国際条約によって保護されています。
このアプリケーションの動作には.NET Framework
OK
CHAMONIX

8. 原点復帰をおこないましょう

8.1. 全軸原点復帰



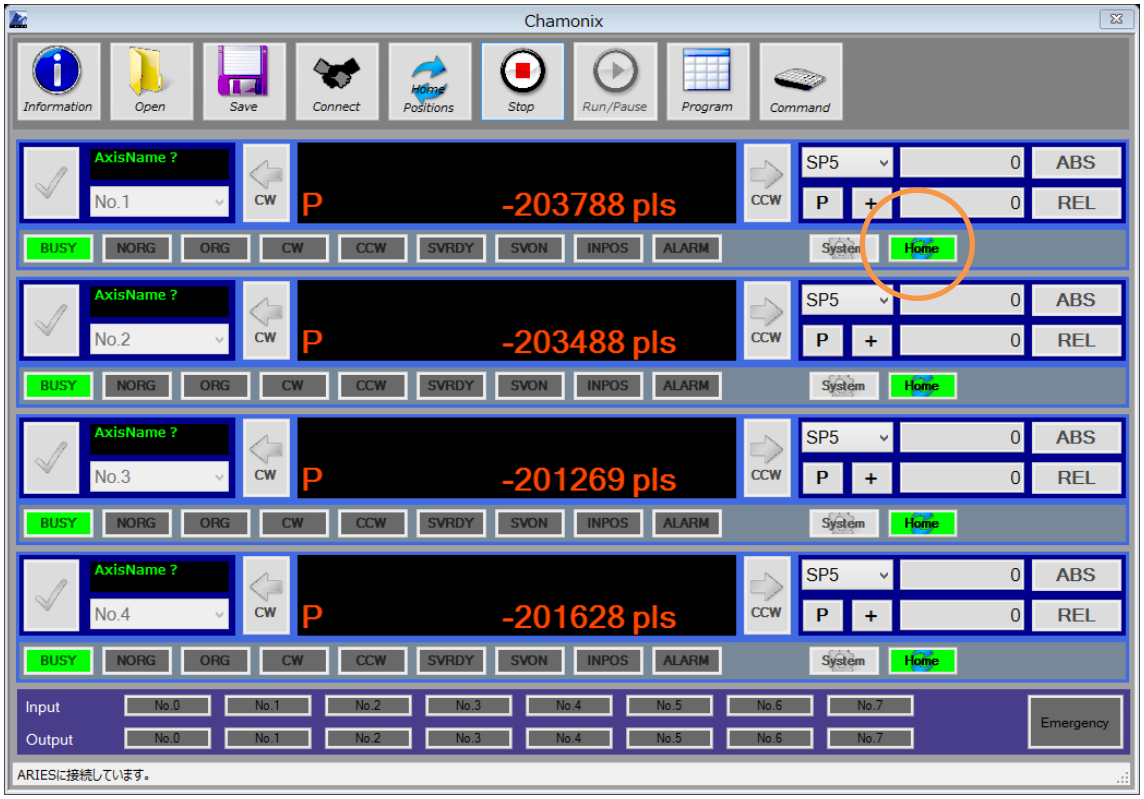
「Home Positions」を押すと、コントローラが制御するすべての軸が原点復帰します。

各軸は他の軸の動きを考慮せず独立して原点復帰動作を行います。それぞれの干渉に十分ご注意ください。

軸が干渉する恐れがある場合には、次項で説明する単軸の原点復帰をご利用ください。原点復帰のパターンはご使用のコントローラの説明書をご覧ください。原点復帰方式はコントローラのシステムパラメータの設定値に従います。システムパラメータの設定方法は後述します。

⚠ 全軸原点復帰を実行されたことにより。もし軸同士が干渉し不具合が生じたとしても、当社として責任は負いかねます。

8.2. 原点復帰中のステータス



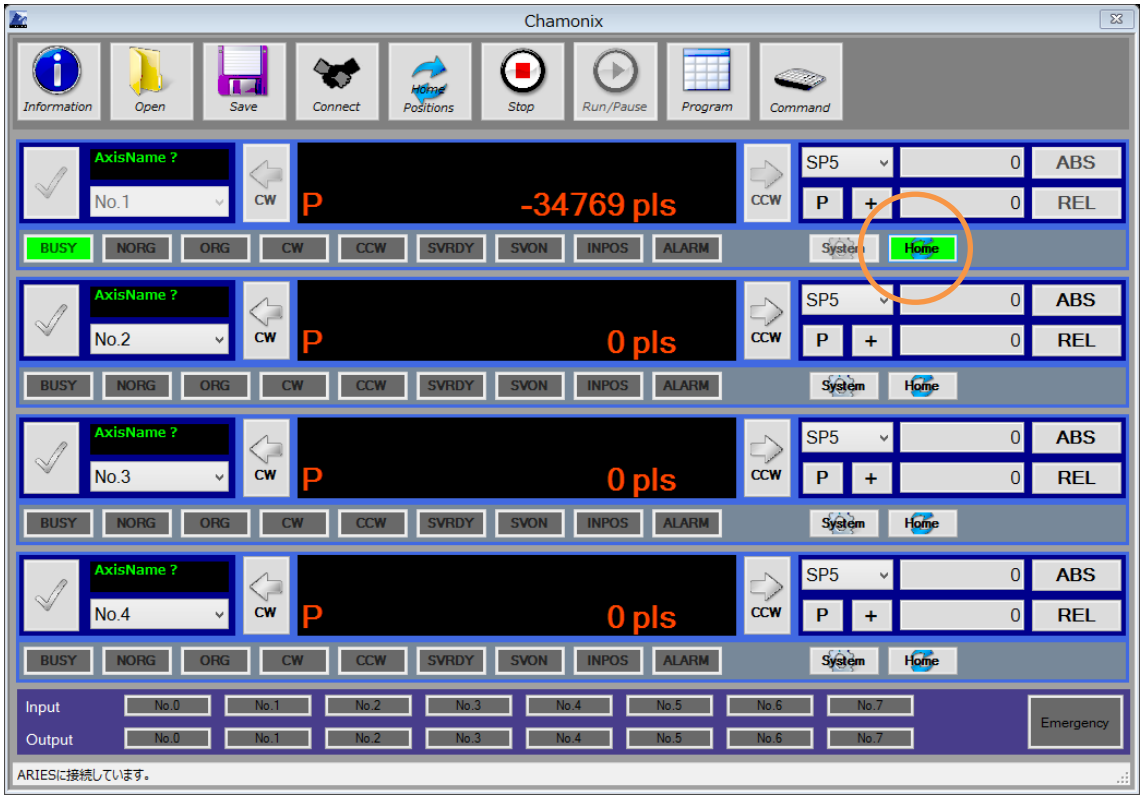
原点復帰が実行されている間、各軸の「Home」ボタンが点灯します。

それぞれの軸の原点復帰が完了すると「Home」ボタンは消灯します。

点灯中の「Home」ボタンをクリックするとその軸だけ原点復帰を中止します。

駆動中は「BUSY」ランプも点灯します。

8.3. 単軸の原点復帰



原点復帰は軸の個別の「Recovery」ボタンをクリックすることで、特定の軸のみの原点復帰をすることも可能です。

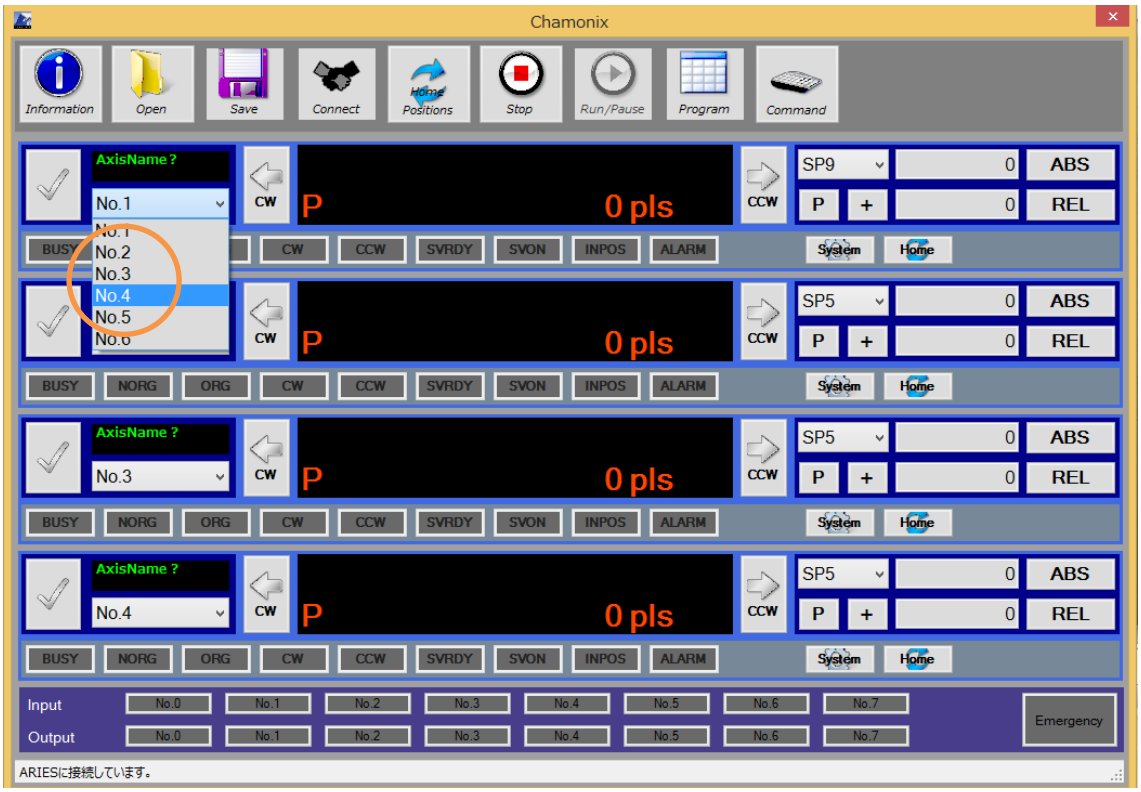
軸同士が干渉する恐れがある場合には、軸毎に一つずつ原点復帰をされることをお勧めします。

点灯中のボタンを押すと停止します。

⚠ 全軸原点復帰を実行されたことにより。もし軸同士が干渉し不具合が生じたとしても、当社として責任は負いかねます。

9. ステージを動かしてみましょう

9.1. 制御する軸の選択

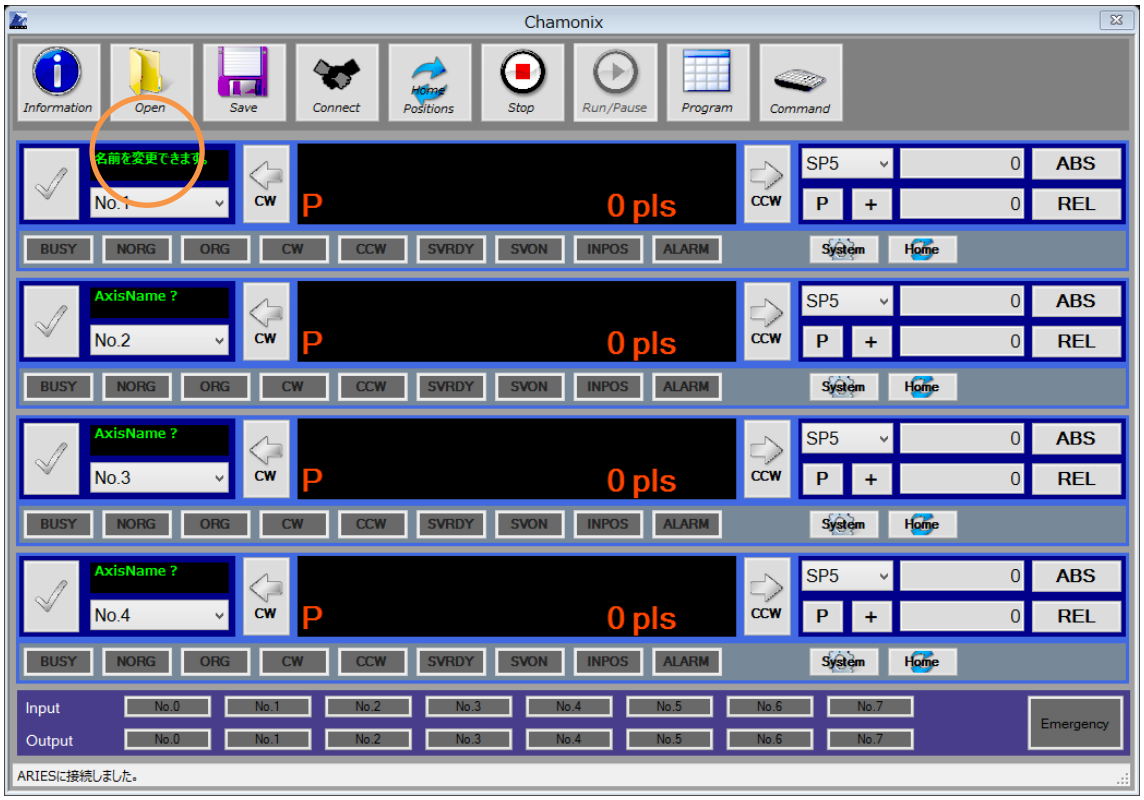


コンボボックスを使用して制御する軸を選択することができます。駆動中に GUI に表示して制御する軸を変更することも可能です。

表示は最大４軸までですが、３２軸まで制御できます。

CRUX の場合は 2 軸までの制御になります。ARIES の場合、LYNX の増設によって最大 32 軸まで制御できます

9.2. 軸の名前の入力

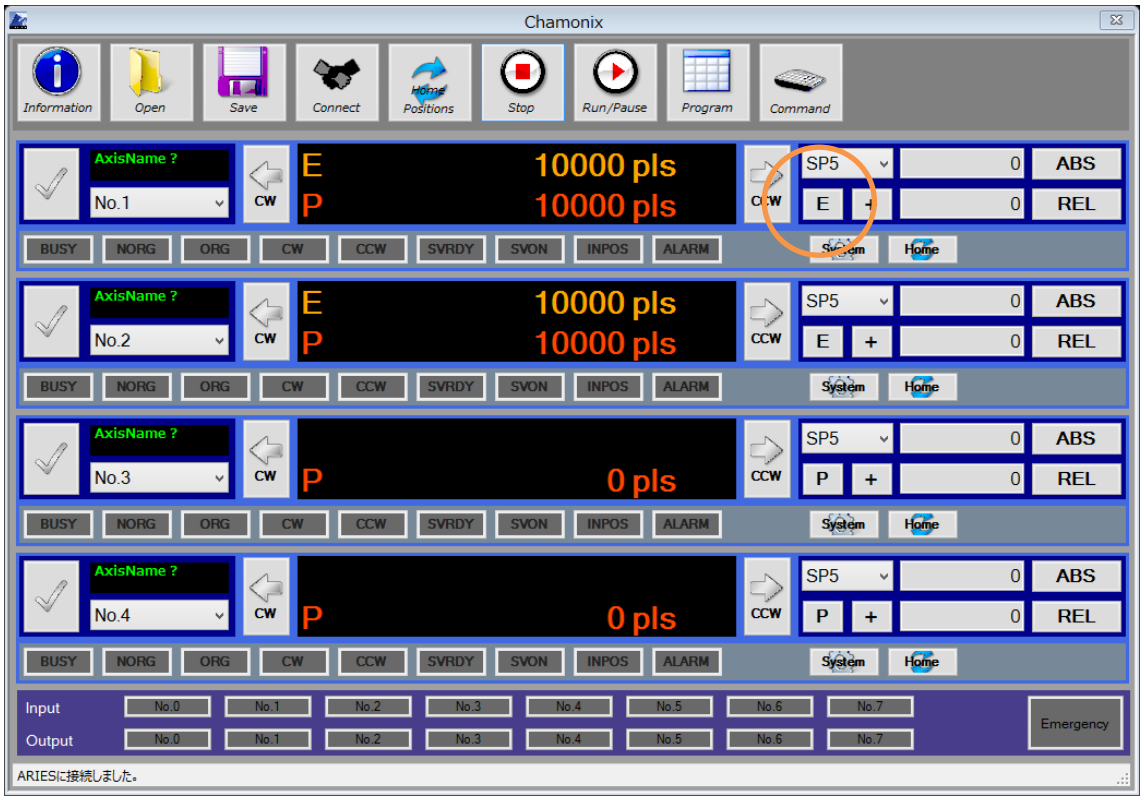


軸の名称を入力できます。

軸の名称は起動ファイルあるいは Config ファイルへ保存されます。

Windows で入力ができる言語で書き込めます。文字コードは UTF-8 です。

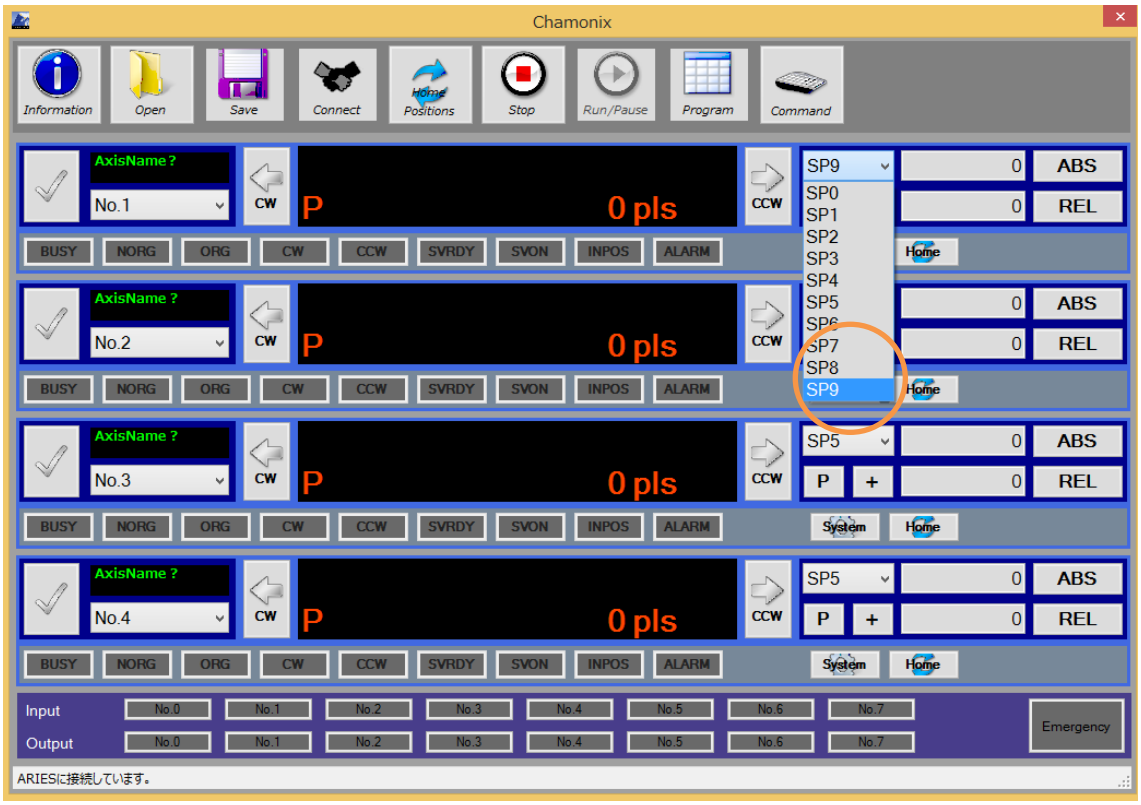
9.3. 現在位置表示の切り替え



「P」ボタンを押下すると表示が「E」へ変わり、エンコーダの現在位置が表示されます。

「E」ボタンを押下すると表示が「P」へ変わり、モータパルスの現在位置が表示されます。

9.4. 速度テーブルの選択

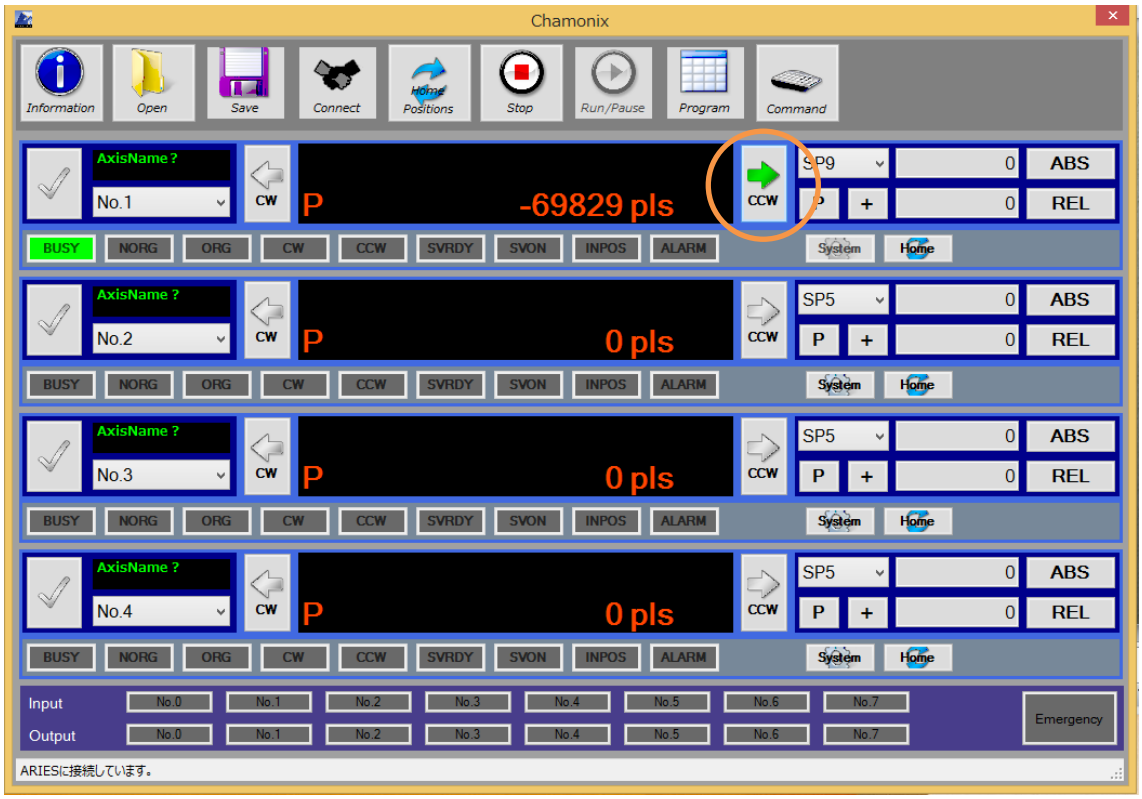


それぞれの軸の速度テーブルはコンボボックスから変更できます。

速度テーブルの設定値はシステムパラメータの変更を行うことで設定できます。システムパラメータの変更方法は 12.2 で説明します。

速度テーブルの各パラメータはコントローラの型式によって異なります。詳しくは各コントローラの取扱説明書をご覧ください。

9.5. マニュアルでの移動

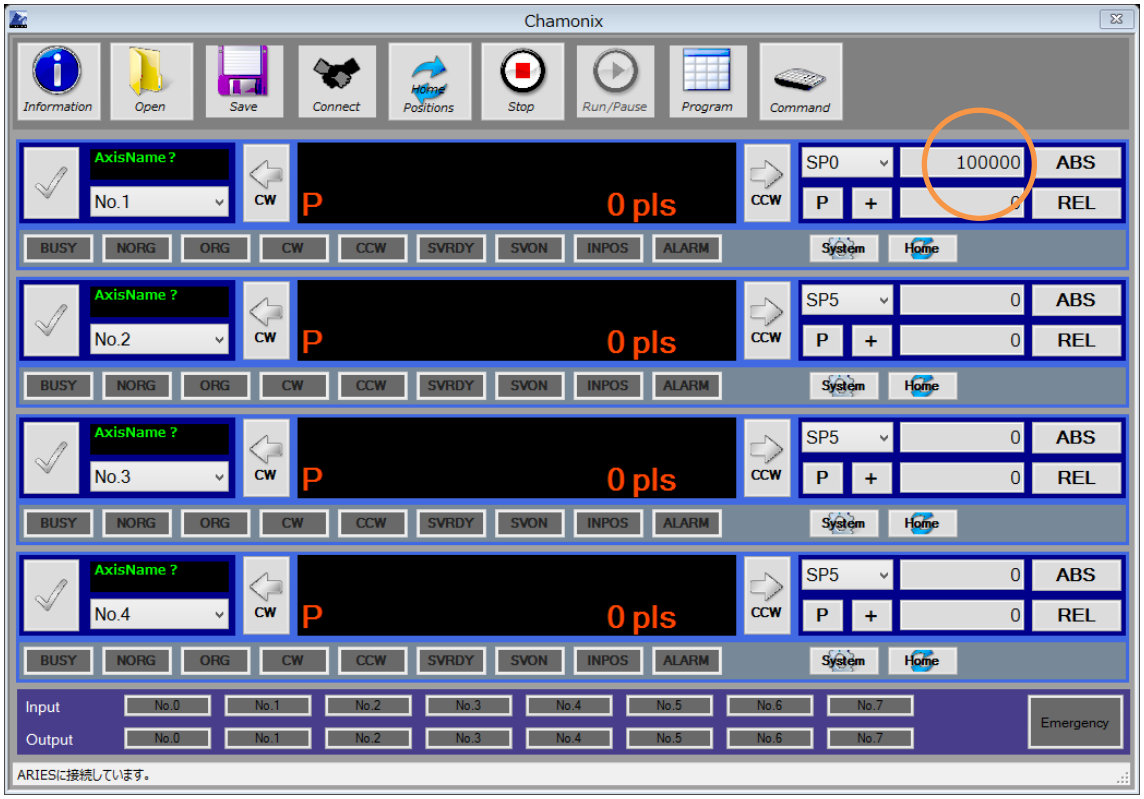


各軸の CW ボタンを押下すると、その間 CW 方向へ移動します。

同様に CCW ボタンを押下すると、押している間 CCW 方向へ移動します。

10. 位置決め

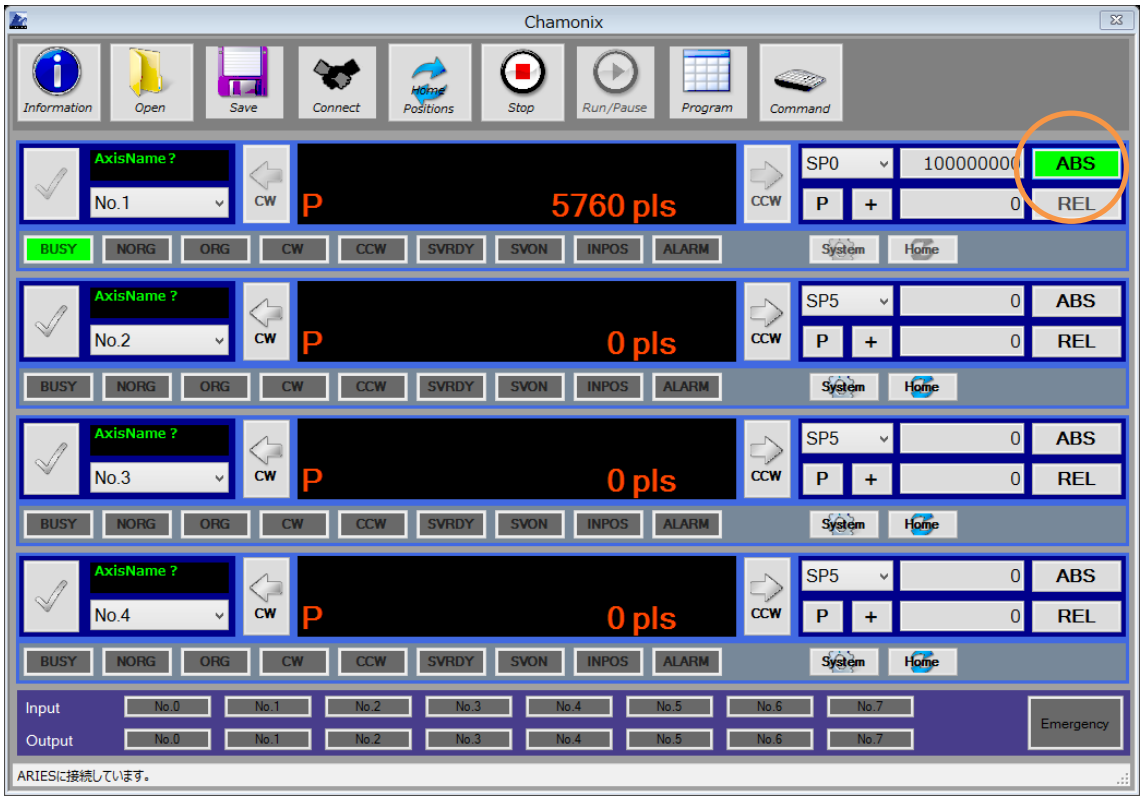
10.1. 絶対位置管理による位置決めの設定



このテキストボックスに移動先の絶対位置を入力してください。

システムパラメータに制御分解能の換算値が設定されている場合、換算値を入力してください。デフォルトではモータパルスを入力する設定になっています。

10.2. 絶対位置管理による位置決めの実行

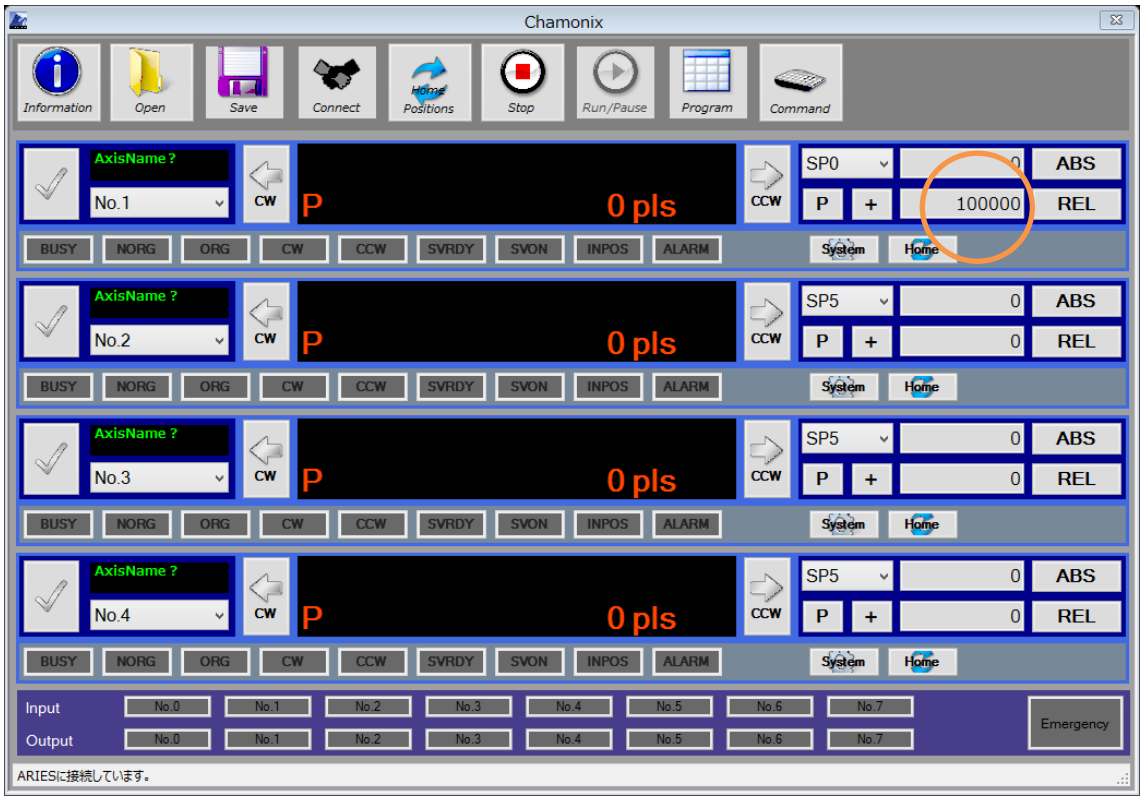


絶対位置を指定し、ABS ボタンをクリックすると、絶対位置管理で位置決めを実行します。

点灯中の「ABS」ボタンを押すとこの軸のみが停止します。

モータ駆動中は BUSY ランプが点灯します。

10.3. 相対位置管理による位置決めの設定

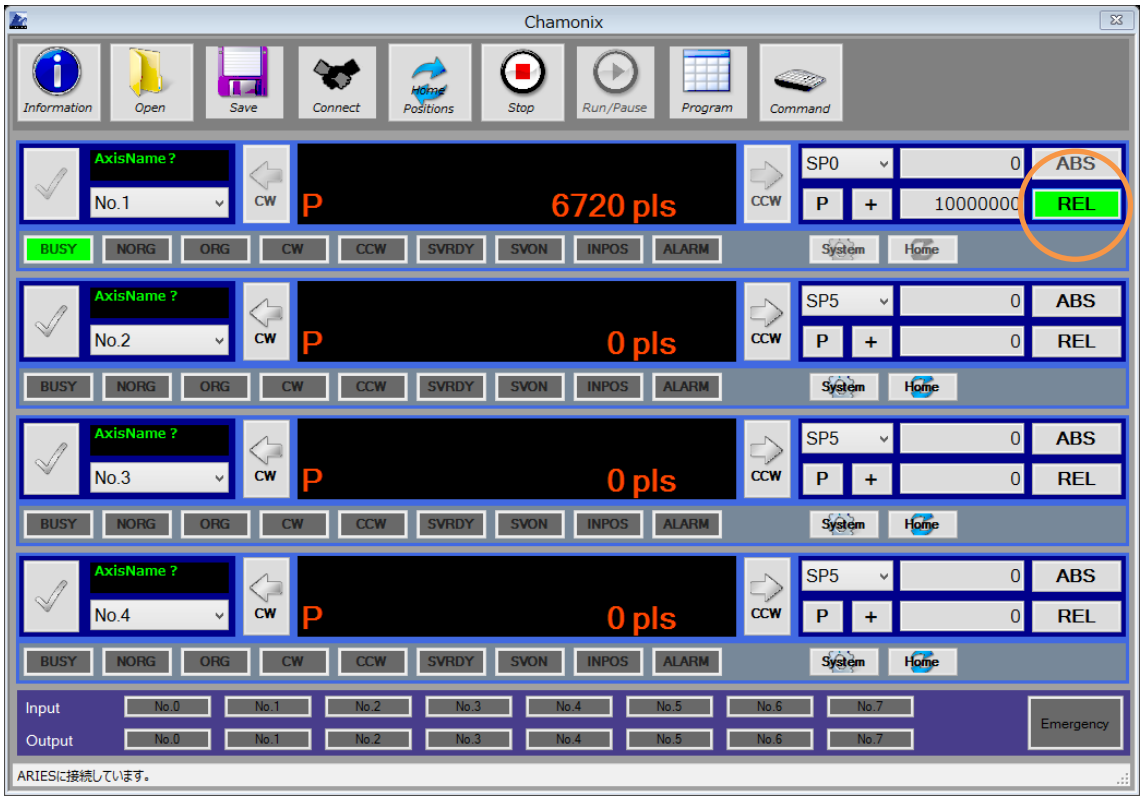


このテキストボックスに移動先の相対移動量を入力してください。

符号のボタンをクリックすると符号を変更できます。

システムパラメータに制御分解能の換算値が設定されている場合、換算値を入力してください。デフォルトではモータパルスを入力する設定になっています。

10.4. 相対位置管理による位置決めの実行

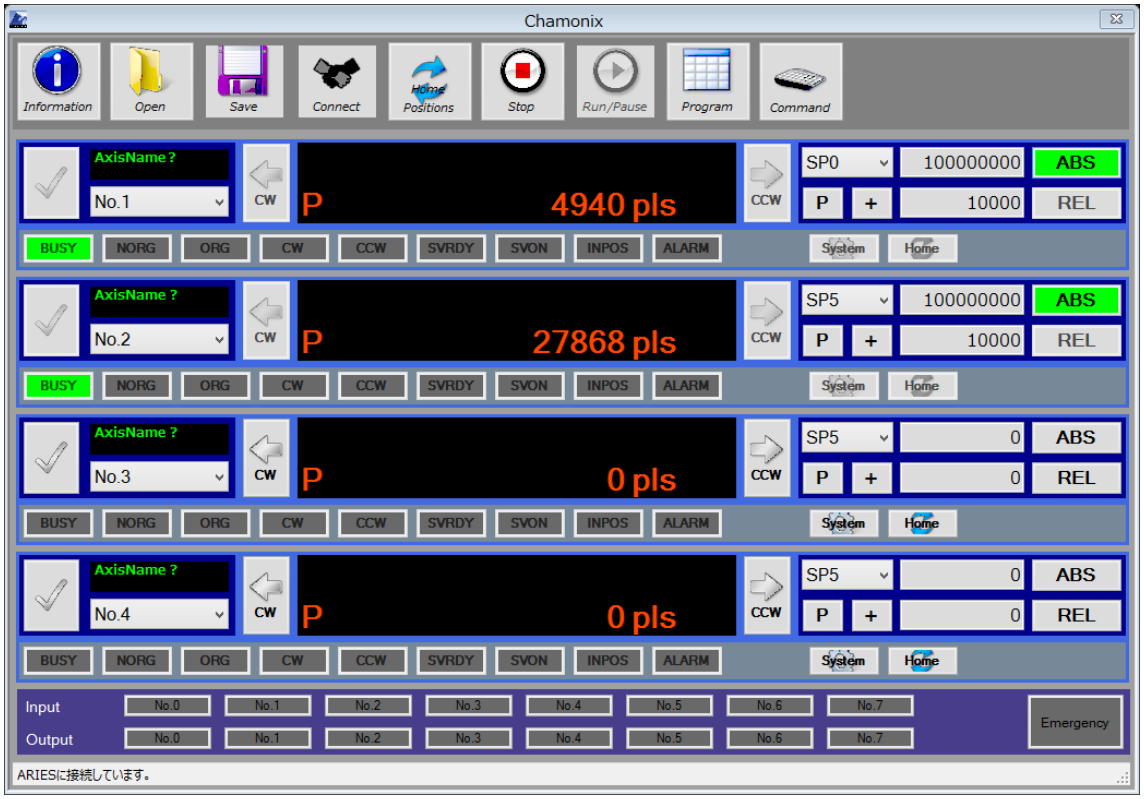


相対位置を指定し、REL ボタンをクリックすると、相対位置管理で移動します。

点灯中の「REL」ボタンを押すとこの軸のみが停止します。

駆動中は BUSY ランプが点灯します。

10.5. 多軸駆動を実行



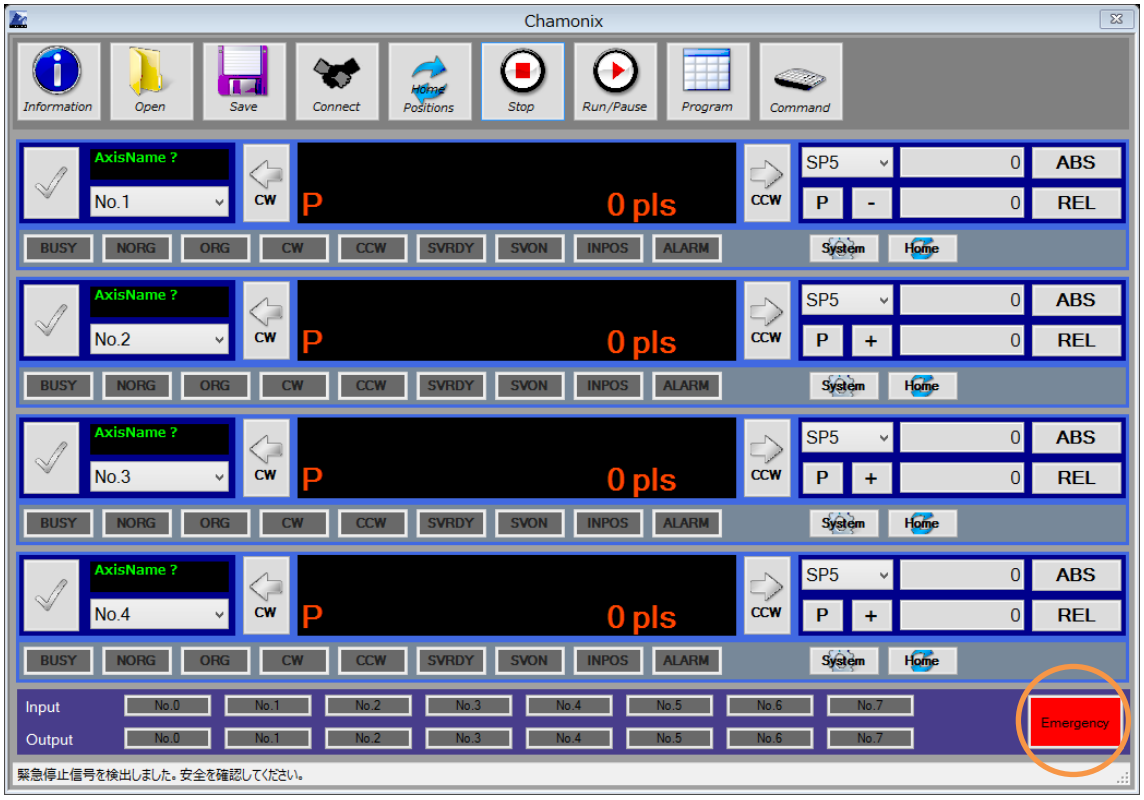
各軸のチェックボタンを選択してから「ABS」あるいは「REL」ボタンを押すと選択された軸を続けて駆動できます。

現在は APS/RPS の駆動命令を順に軸毎に送るのみの仕様です[12]。

[12]今後、バージョンアップによってさまざまな駆動に対応していく予定です。

11. 非常停止を受信

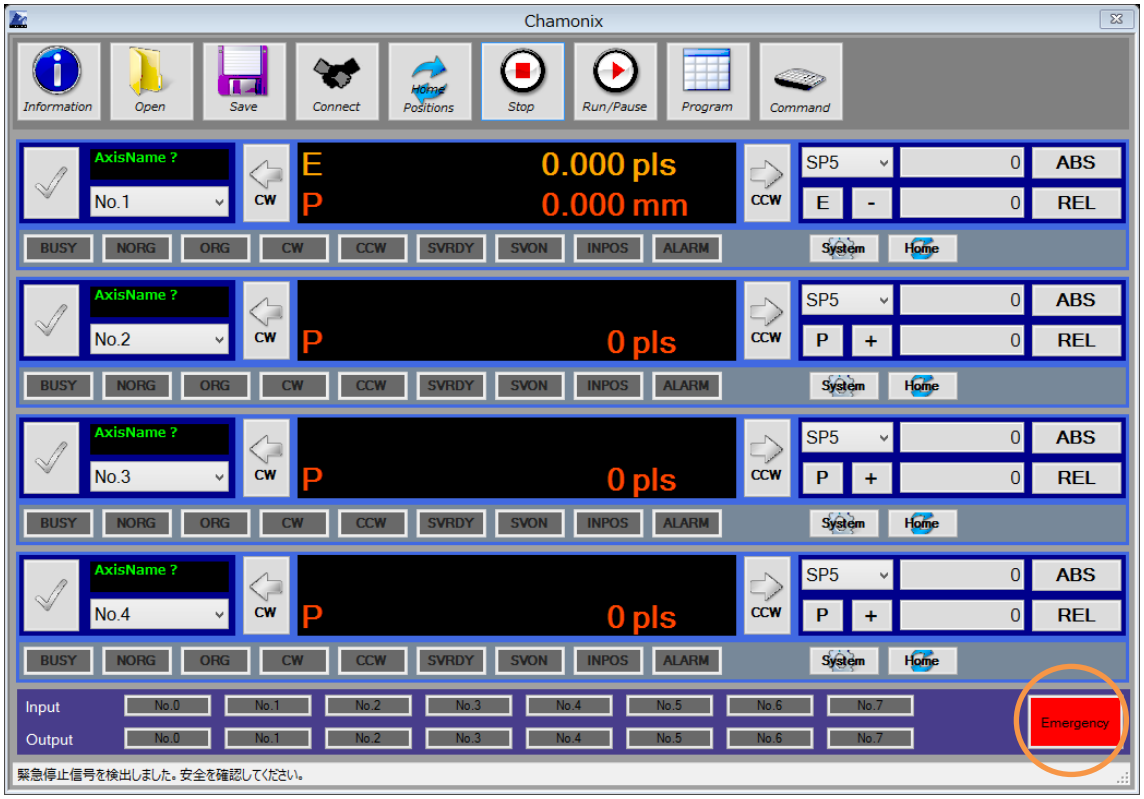
11.1. 非常停止信号を受信（ARIES）



コントローラに ARIES を利用している場合、非常停止信号を受信すると「Emergency」が赤く点灯します。

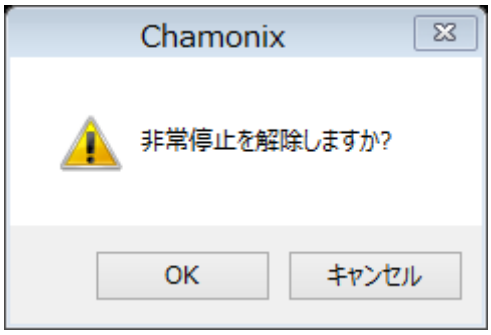
この状態ではコントローラはモータパルスを一切発生しません。このため各モータは駆動できません。

11.2. 非常停止からの復帰（ARIES）



ARIESをご利用の場合、非常停止信号の回路を復帰し、安全を確認した後、「Emergency」ボタンを押すと非常停止が解除できます。

「Emergency」を押下すると下図のメッセージが表示されます。「OK」を押すと非常停止が解除されます。「キャンセル」を押すと非常停止は継続します。



非常停止から復帰する際、非常停止の原因が取り除かれ、安全が確保されていることを十分に確認してください。

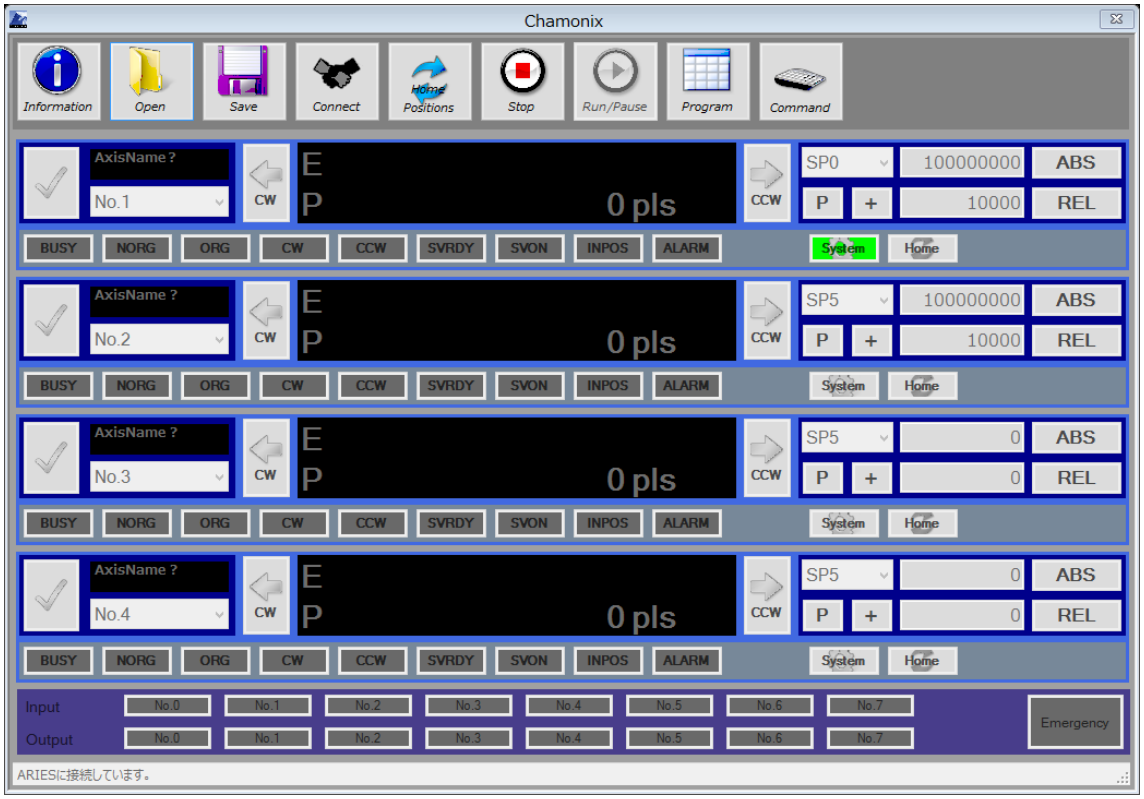
11.3. CRUX の場合

CRUX の場合には、非常停止信号を Chamonix 側で表示することはありません。CRUX のフロントパネルで「EMG」が点灯している場合、非常停止が働いています。

非常停止信号の回路を復帰してください。

12. システムパラメータを変更する

12.1. パラメータ設定を起動する



各駆動軸の「System」ボタンをクリックするとシステムパラメータの設定ウィンドウが「System Parameters」起動します。起動時にコントローラからシステムパラメータを受信します。各軸のシステムパラメータは次頁に示したプロパティとして開きます。

設定できるパラメータはシステムパラメータと速度テーブルです。これらのパラメータはできる限りコントローラ間の互換性を維持するように設計されています[13]。

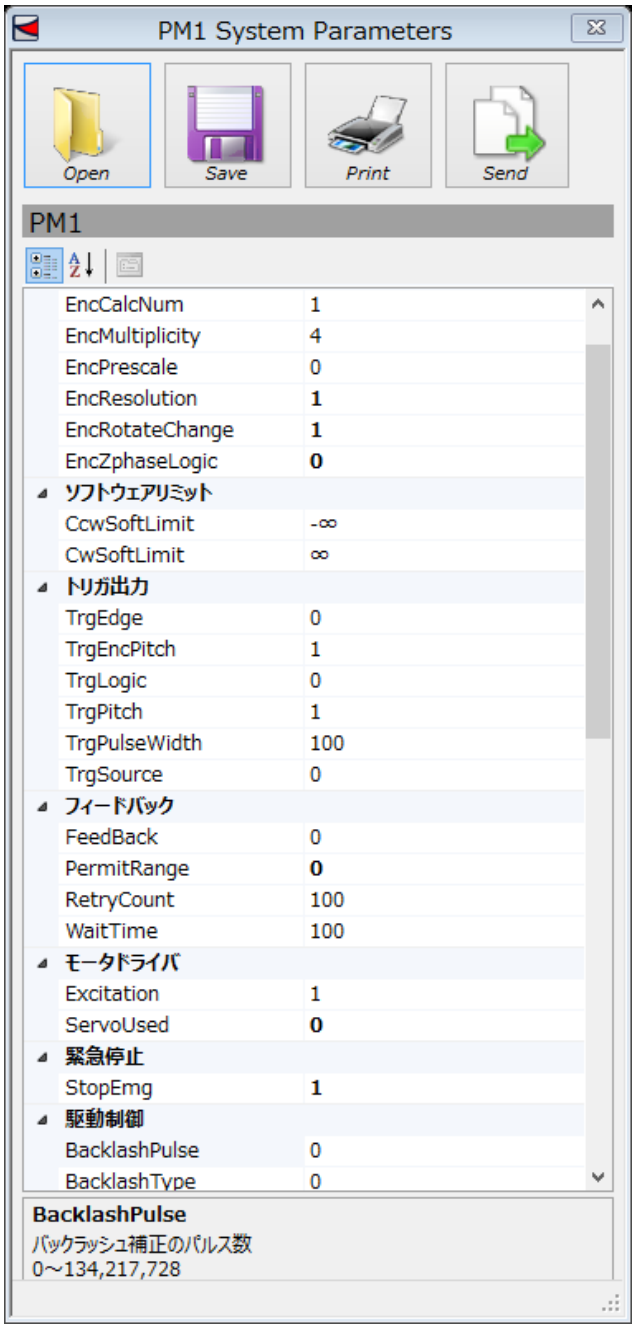
接続されているコントローラで欠けている機能は表には記載されません。保存されたファイルには、

[13]異なる機種で完全に互換性を保証するという意味ではありません。

Chamonix のデフォルト値が保存されます[14]。

[14] 上位機種において、下位機種で保存されたファイルを読み込んだ場合には、ファイルに保存されていないパラメータは Chamonix のデフォルト値で起動されます。

12.2. パラメータを設定する



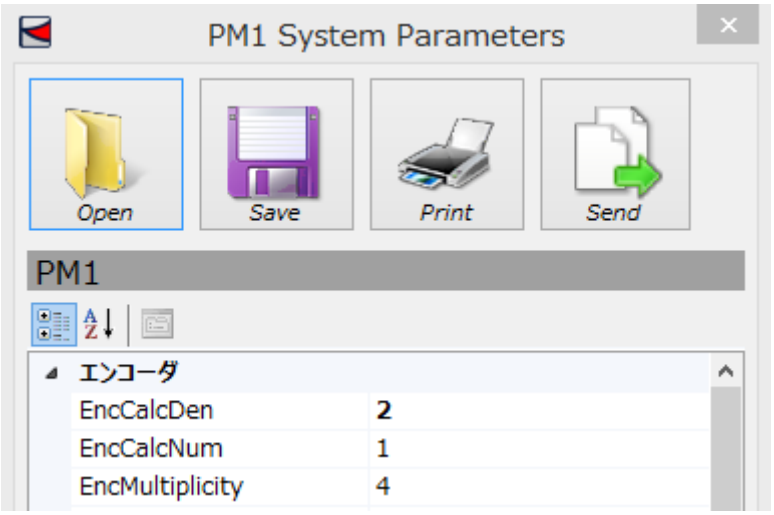
「System　Parameters」には、コントローラのシステムパラメータの値を入力することにより変更できます。

値の変更は「Send」ボタンを押すと一括してコントローラに反映されます。

アプリケーションのプロパティと一体化することでコントローラの機種ごとの番号の違いを吸収しています。

コントローラのデフォルト値が設定されているときプロパティの値はノーマルで、デフォルト値と異なる値が設定されているとき値はボールドで表示されます。

12.3. システムパラメータのファイルへの保存



「Save」ボタンを押下するとシステムパラメータは軸毎にファイルに保存できます。

保存したファイルはこのウィンドウに「Open」ボタンを押下することで読み込むことが可能です。

ファイル形式は、bin ファイル、XML ファイルです。これらのファイルは SC-210/410、CRUX、ARIES で共通に利用できるように設計されています[15]。

接続されているコントローラで欠けている機能は表には記載されません。保存されるときにはデフォルト値が保存されます。

上位機種において、下位機種で保存されたファイルを読み込んだ場合には、ファイルに保存されていないパラメータは Chamonix のデフォルト値で起動されます。

[15]異なる機種で完全に互換性を保証するという意味ではありません。

12.4. パラメータの印刷

Date: 2014/06/09 9:06:50
Controller Type: ARIES

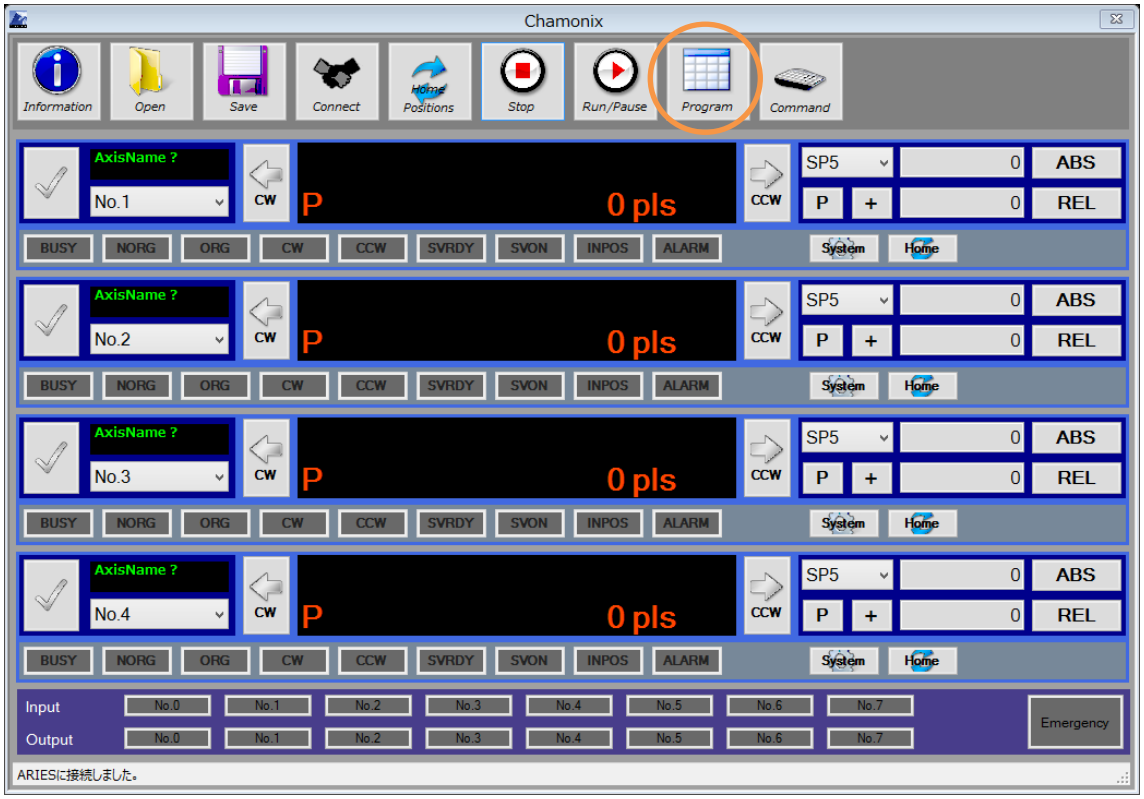
PM1

エンコーダ	
EncCalcDen	**2**
EncCalcNum	1
EncMultiplicity	4
EncPrescale	0
EncResolution	**1**
EncRotateChange	**1**
EncZphaseLogic	**0**
ソフトウェアリミット	
CcwSoftLimit	-∞
CwSoftLimit	∞
トリガ出力	
TrgEdge	0
TrgEncPitch	1
TrgLogic	0
TrgPitch	1
TrgPulseWidth	100
TrgSource	0
フィードバック	
FeedBack	0
PermitRange	**0**
RetryCount	100
WaitTime	100
モータドライバ	
Excitation	1
ServoUsed	**0**
緊急停止	
StopEmg	**1**
駆動制御	
BacklashPulse	0
BacklashType	0
LimitSwap	0
PmClock	2
PmLogic	0
PmPrescale	0
PmRotateChange	0
Resolution	**1**
TopSpeedLimit	**50000**
原点復帰	
EncSyncWrite	**1**
OrgOffset	0
OrgScanSpeed	500
OrgType	4
信号論理	
LimitLogic	**1**
NOrgSignalLogic	0
OrgSignalLogic	0
速度テーブル	
Speed_Hi	**10,8000,50,15,3**
Speed_Lo	**5,4000,25,10,3**
SpeedNo0	**500,1000,16,16,2**
SpeedNo1	**500,2000,20,20,2**
SpeedNo2	**500,3000,24,24,2**
SpeedNo3	**500,4000,28,28,2**
SpeedNo4	**500,5000,32,32,2**
SpeedNo5	**500,6000,36,36,2**
SpeedNo6	**500,7000,40,40,2**
SpeedNo7	**500,8000,44,44,2**
SpeedNo8	**500,9000,48,48,2**
SpeedNo9	**500,10000,52,52,2**

「PRINT」ボタンをクリックするとパラメータを印刷できます。

13. プログラマブルグリッド

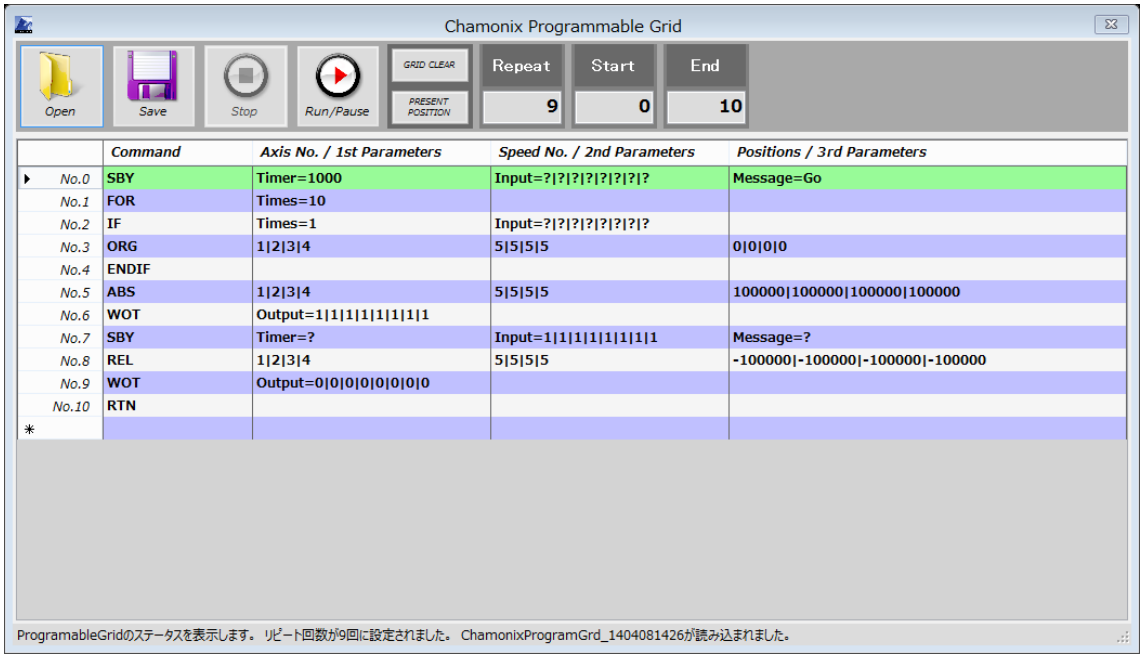
13.1. プログラマブルグリッドの起動



「Program」ボタンをクリックすると「Programmable Grid」が起動します。

「Programmable Grid」はこれまでの Customer の Setting Grid を進化させ、簡単なマクロのような機能まで付加したティーチング機能です。

13.2. プログラマブルグリッドの基本機能

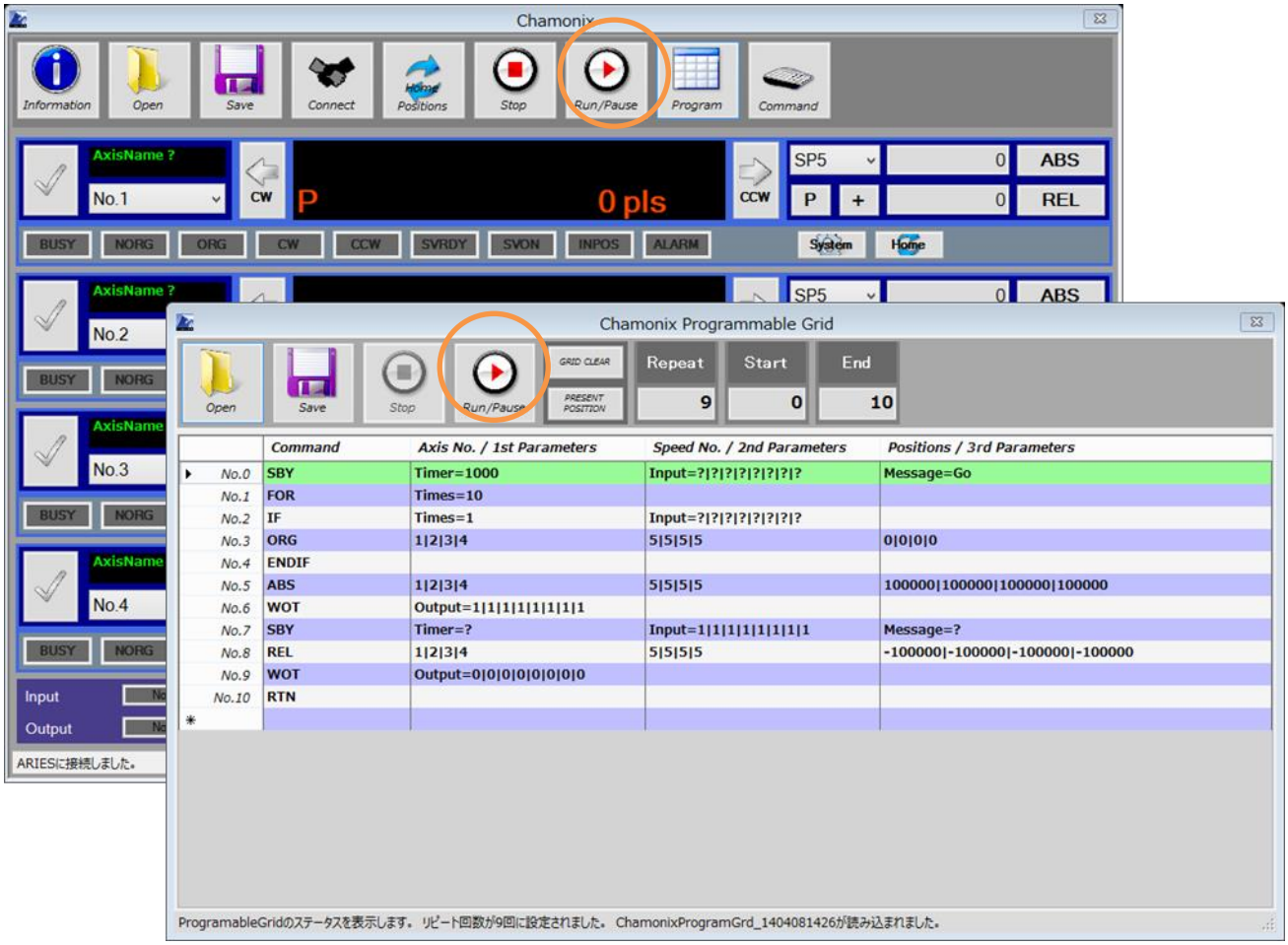


グリッド内に記述したプログラムに従ってコントローラに接続されたステージを自動駆動します。

グリッドの行に記述されたコマンドは原則として上から順番に実行されます。

Start で指定された行と End で指定された行を Repeat で指定された回数繰り返します。Repeat に「Endless]あるいは負の数字を入力すると無限に繰り返します

13.3. プログラムを実行

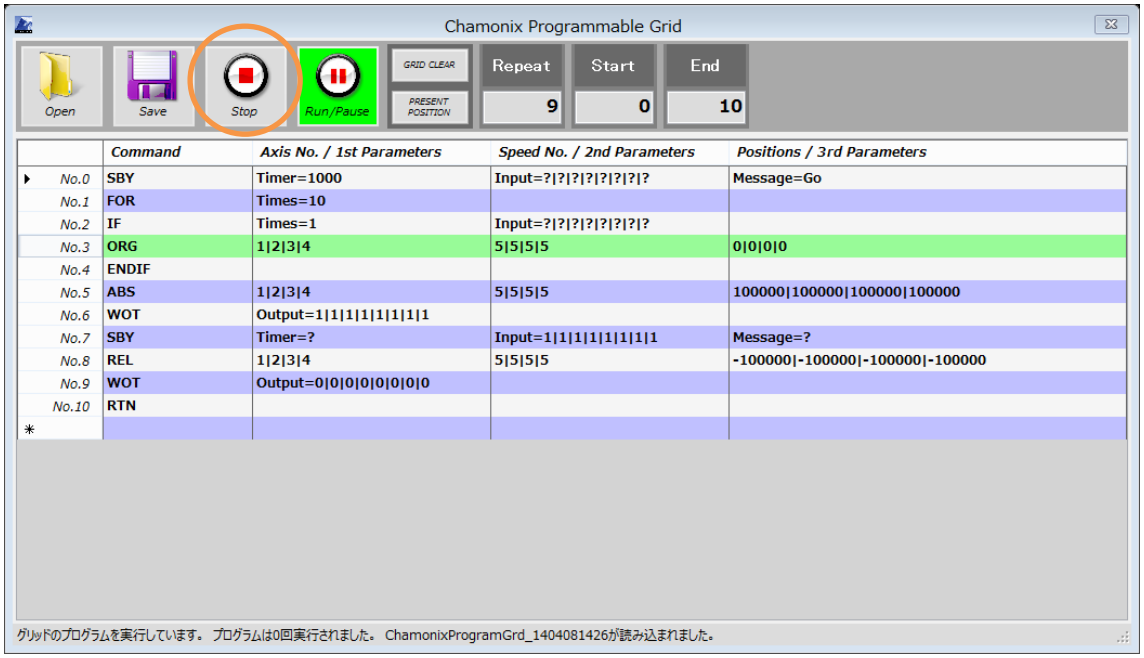


「Chamonix」あるいは「Chamonix ProgramableGrid」の「RUN」ボタンをクリックすると実行します。

実行中は「RUN」ボタンは「PAUSE」ボタンになり、ランプが点灯します。実行中の行は緑色に点灯します。

「PAUSE]ボタンをクリックすると実行中の行を実行したのちに一時停止します。一時停止からの再開は再び「RUN」ボタンを押して下さい

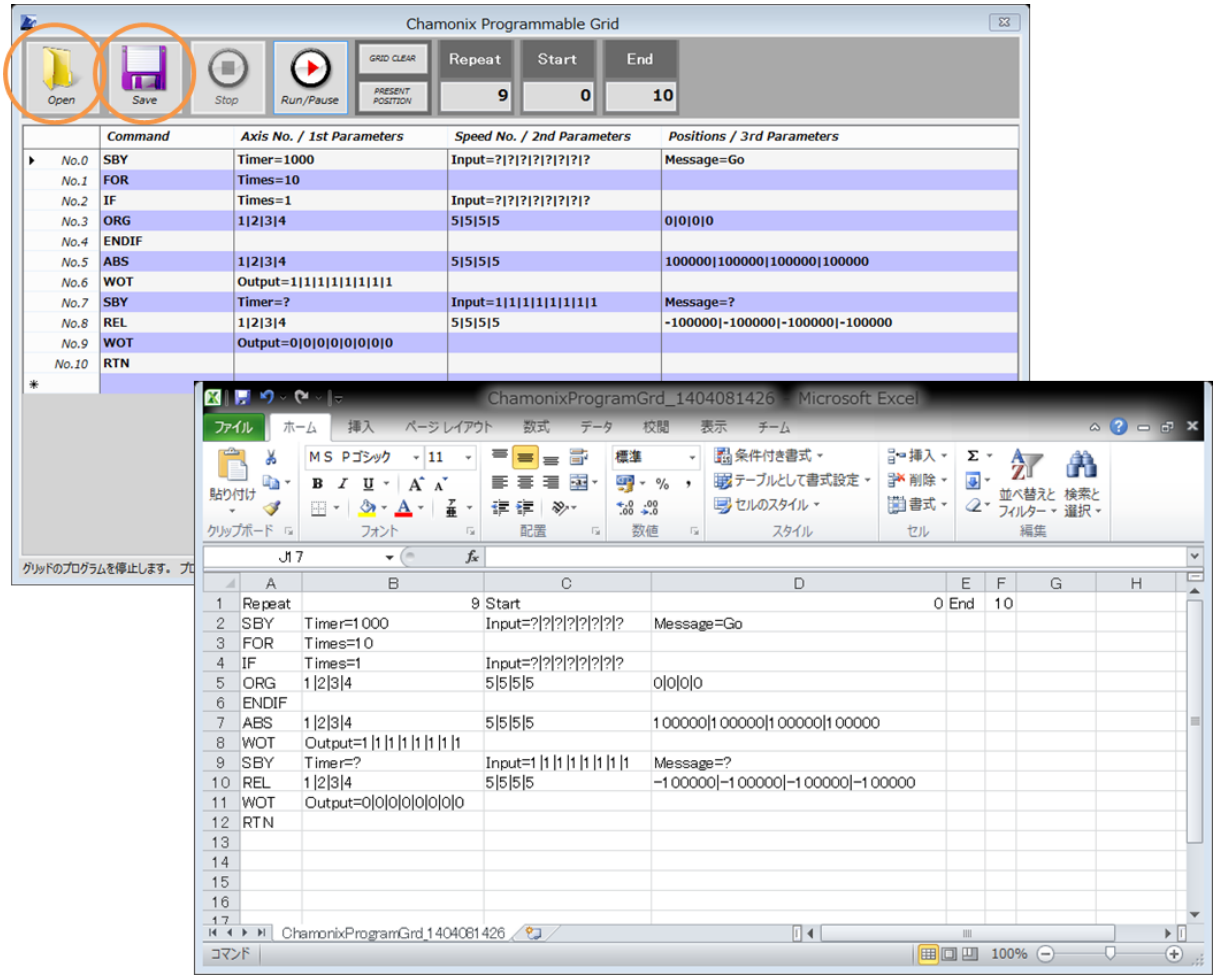
13.4. プログラムの停止



「STOP」ボタンをクリックすると停止します。「STOP」ボタンによる停止は緊急停止を意味し、直ちに停止します。

停止位置からの再開はできません。再開する可能性がある場合は「Pause」ボタンをご利用ください。

13.5. プログラムデータの入出力



プログラムデータはファイルに保存することができます。

データはファイルから読み込むことも可能です。

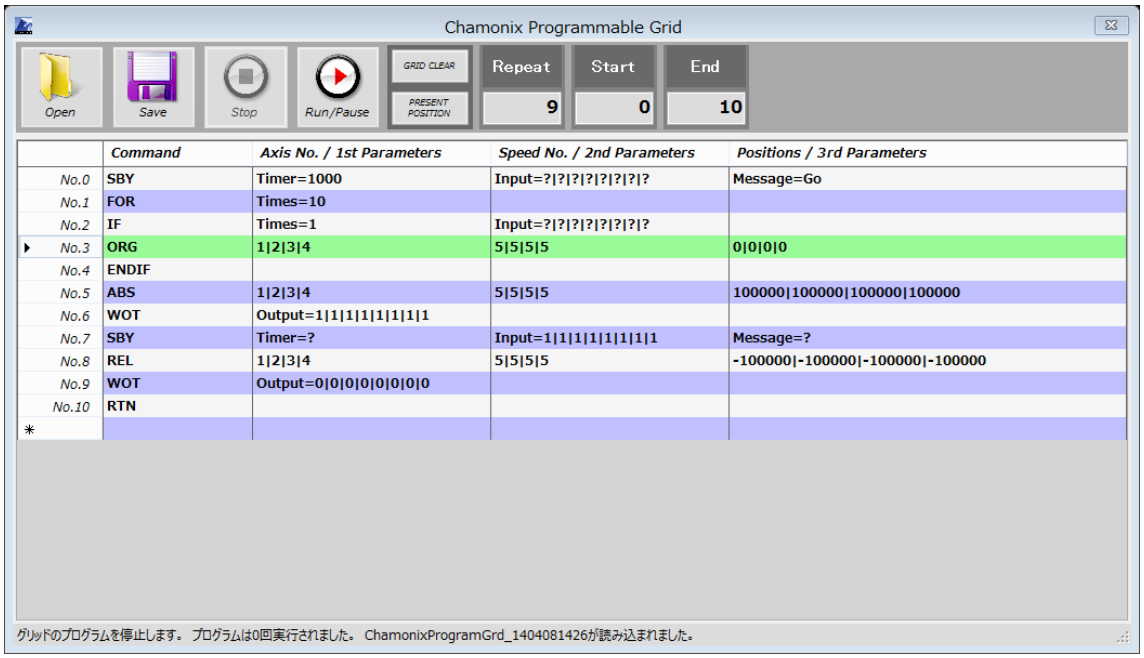
ファイルは DAT（標準）、XML、CSV（CSV1)の形式に対応しています。DAT の形式を推奨しています。

Parameters 内の各パラメータの区切り文字は「|」あるいは「/」を利用します。内部処理の際に「|」に読み替えられます。これは CSV を Microsoft Excel で処理する場合の不都合[16]を考慮してのものです。

[16] Excel の場合、デフォルトの設定において、「1/2」は 1 月 2 日と自動的に日付関数に変換されてしまうためです

13.6. プログラマブルグリッドのコマンド

13.6.1. ORG: 原点復帰



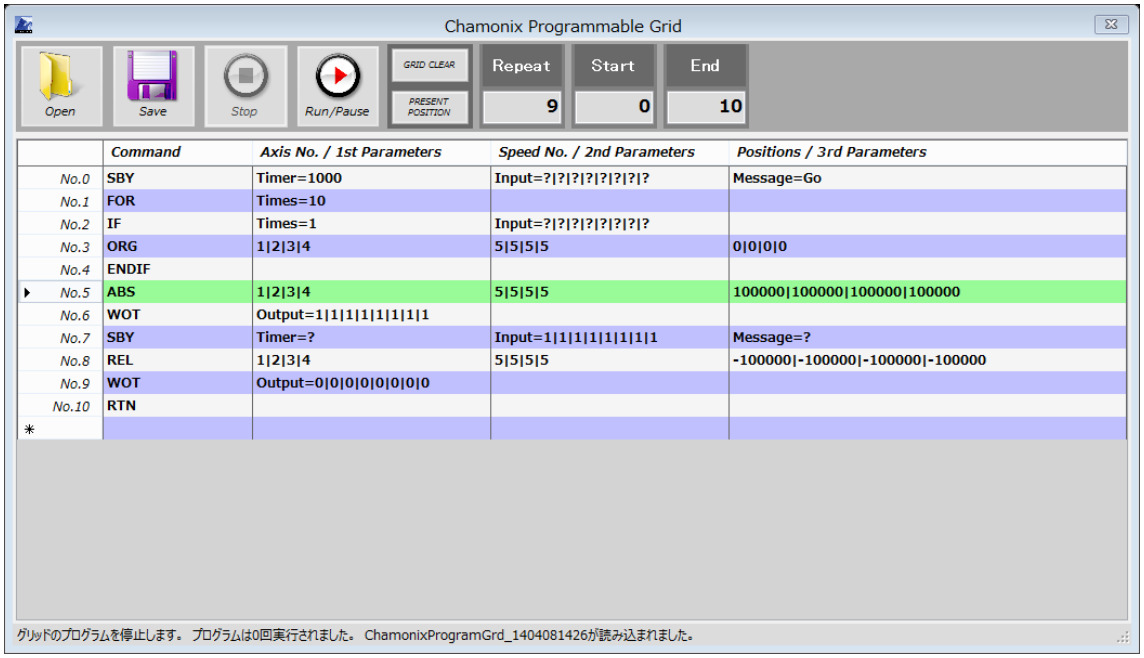
ORG はシステムパラメータの設定に従って原点復帰動作を行います。

原点復帰動作完了後、
1st Parameter で指定された軸を、
2nd Parameters で指定した速度テーブルで、
3rd Parameters で指定された絶対位置へ移動します。

システムパラメータ設定で「Resolution」プロパティを設定している場合、換算値で指定して下さい[17]。

[17] デフォルトではパルスで入力するように設定されています。

13.6.2. ABS： 絶対位置管理による位置決め

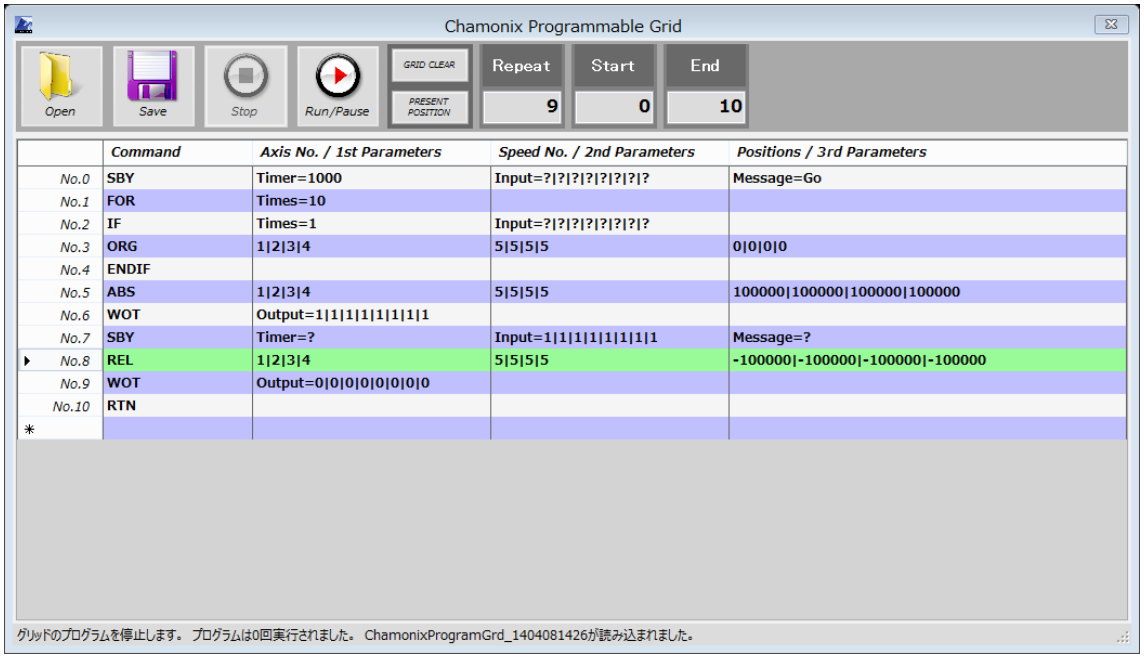


ABS は絶対位置管理で
1st Parameters で指定した軸に
2nd Parameters で指定したスピードで、
3rd Parameters で指定した位置へ移動します。

システムパラメータ設定で「Resolution」プロパティを設定している場合、換算値で指定して下さい[18]。

[18] デフォルトではパルスで入力するように設定されています。

13.6.3. REL: 相対位置管理による位置決め

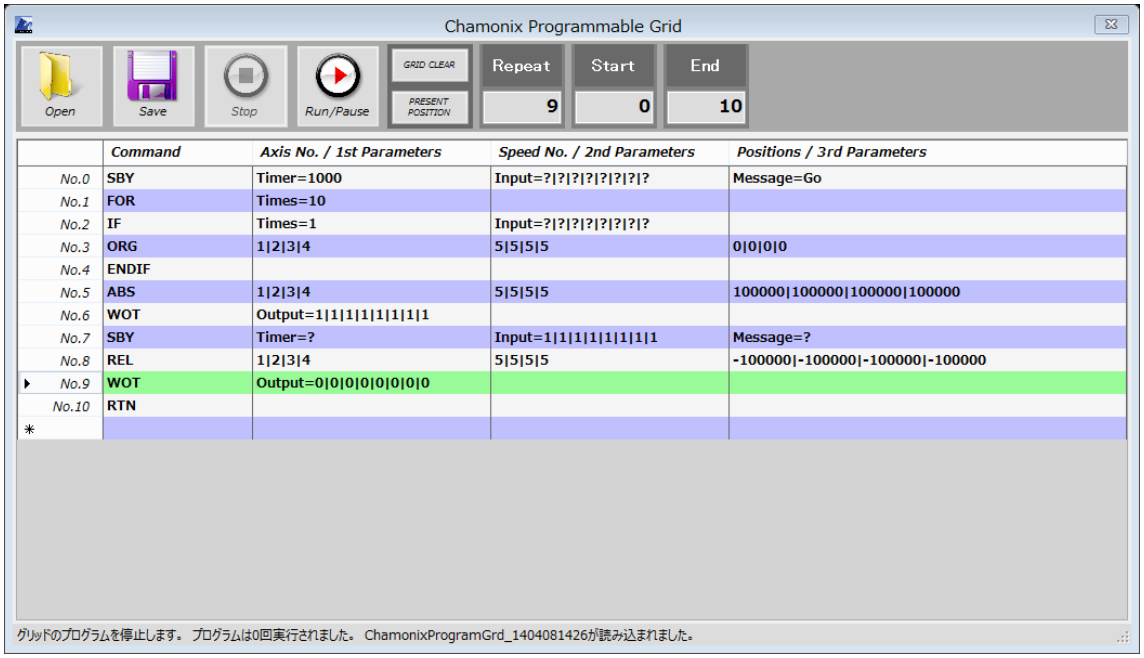


REL は相対位置管理で
1st Parameters で指定した軸に
2nd Parameters で指定したスピードで、
3rd Parameters で指定した位置へ移動します。

システムパラメータ設定で「Resolution」プロパティを設定している場合、換算値で指定して下さい[19]。

[19] デフォルトではパルスで入力するように設定されています。

13.6.4. WOT： 汎用出力の制御



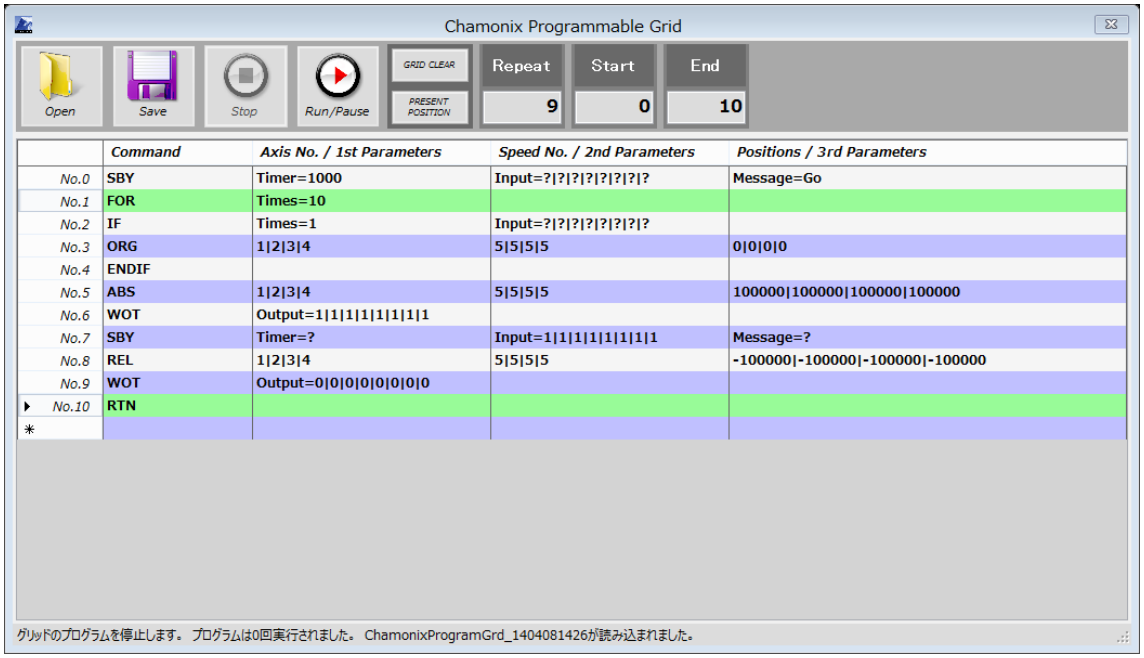
ARIES に対しては I/O の出力を制御することが可能です。上の例では ARIES の 8ch の出力全てを ON にすることを示しています。

1st Parameters を Output=0|0|0|0|0|0|0|0 に設定した場合には、出力全てを OFF にすることを示します。

?はパラメータが Null であることを示します。

I/O を備えていないコントローラが接続されているとき、このコマンドは無視されます。

13.6.5. FOR： 繰り返しループ



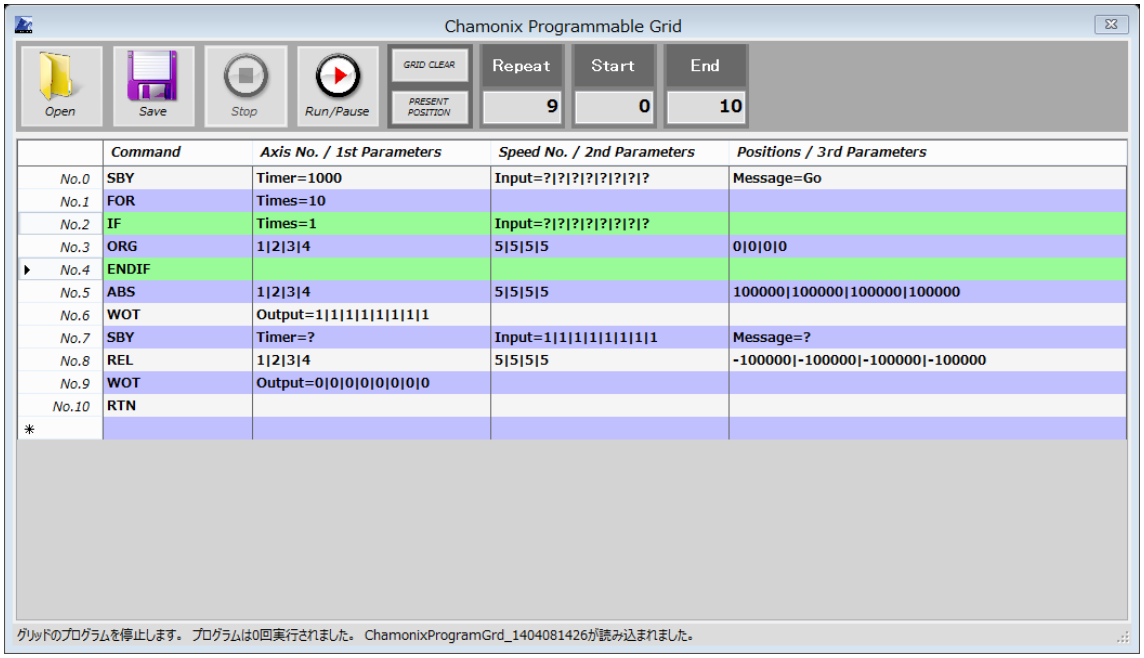
FOR コマンドは、これより下の行で最初に現れる RTN コマンドまでの間、
1st Parameters の「Times=」パラメータ指定子で指定された回数を繰り返します。

FOR コマンドは入れ子にすることができます。FOR コマンドが入れ子にされたとき、内側の FOR コマンドのループから実行されます。

IF コマンドの「Times=」パラメータ指定子も実行中のループの回数を対象とします。

FOR コマンドを指定した場合には、FOR コマンドひとつに対して、FOR コマンドより下側に必ず RTN コマンドを挿入する必要があります。

13.6.6. IF： 条件付きの実行



IF コマンドは 1st Parameters または 2nd Parameters に書かれた条件を満たした場合に、ENDIF コマンドまでの行を実行します。

1st Parameters には、Times=で指定したループの回数を指定できます。上の例ではループが 1 回目の場合のみ ORG コマンド（原点復帰）が実行されます。

2nd Parameters には、コントローラの I/O の入力ステータスを条件として指定できます。I/O を持たないコントローラでは無視されます。

「?」を入力するとその項目は無視されます。

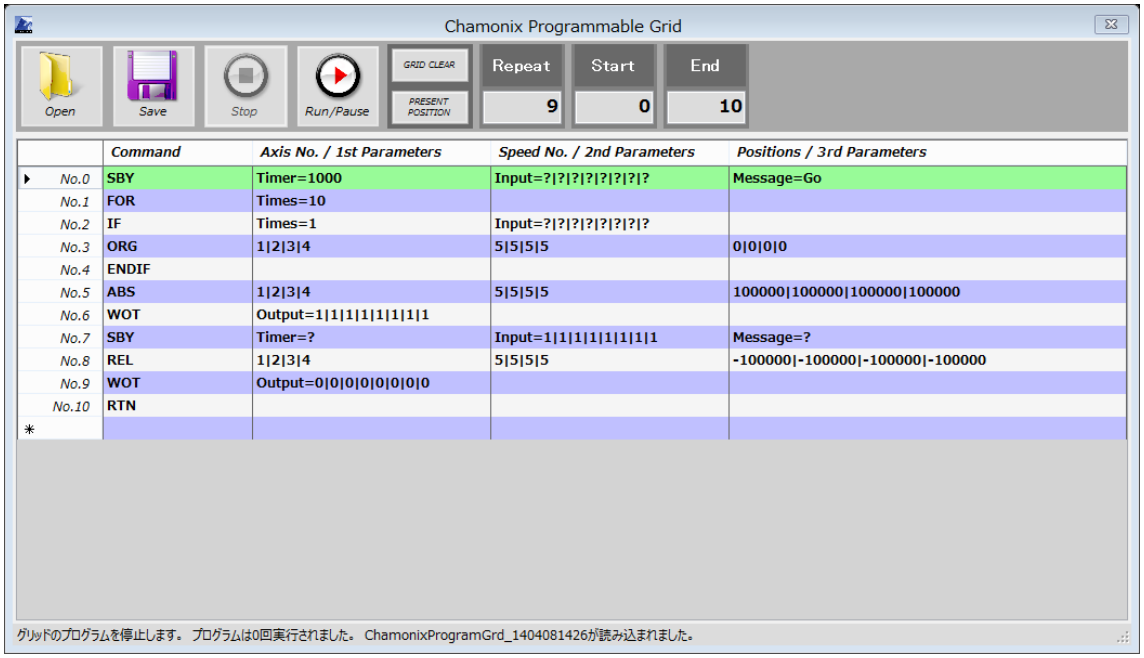
上の例では No.1 から No.10 まで 10 回繰り返すループで、最初の 1 回目のときだけ No.3 の行が実行されるように設定されています。

13.6.7. GOTO： 行のジャンプ

GOTO コマンドは 1st Parameters で指定された行にジャンプします。

GOTOコマンドは大変便利なコマンドですが、注意深く使用しないと、プログラムが複雑になり、間違ったプログラムを設定してしまう恐れがあります。

13.6.8. SBY： スタンバイ



SBY コマンドは 1st から 3rd までの Parameters のいずれかの条件を満たすまで待機します。

1st Parameters の「Timer=」のパラメータ指定子はミリ秒単位でタイマーを指定できます。
2nd Parameters の「Input=」の指定子は ARIES の I/O の入力を指定できます。
3rd Parameters の「Message=」のパラメータ指定子は指定された文字列が WindowsAPI の SendMessage を利用して Chamonix のメインウィンドウへ送信されたときに停止から抜けます。

「?」はそのパラメータを指定しないことを意味します。

13.6.9. プログラマブルグリッドのコマンド一覧

Programmable Grid のコマンドをまとめます。

詳しくは各コマンドの説明を参照してください

ABS

- 絶対位置管理による位置決めを伴う駆動を行います

REL

- 相対位置管理による位置決めを伴う駆動を行います

ORG

- 原点復帰を行い、原点復帰後に絶対値管理で位置決めを行います

SBY

- 条件を満たすまで待機します

FOR

- RTNで指定された行まで、Times指定子で指定された回数を繰り返します

RTN

- FORループの折り返し行を指定します

IF

- ENDIFで指定された行までを条件を満たした場合に実行します

ENDIF

- IFコマンドで条件付き実行をする範囲の終端を意味します

WOT

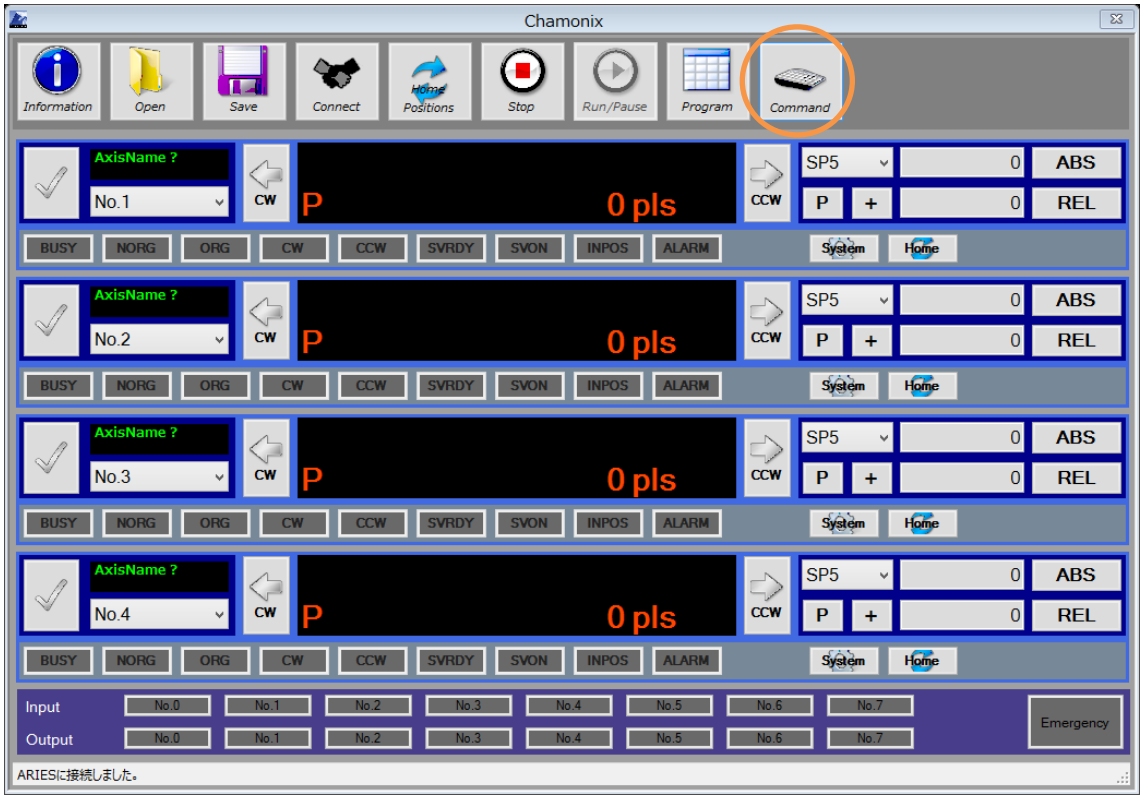
- I/O出力を設定します

GOTO

- 指定した行にジャンプします

14. 直接コマンドを入力する

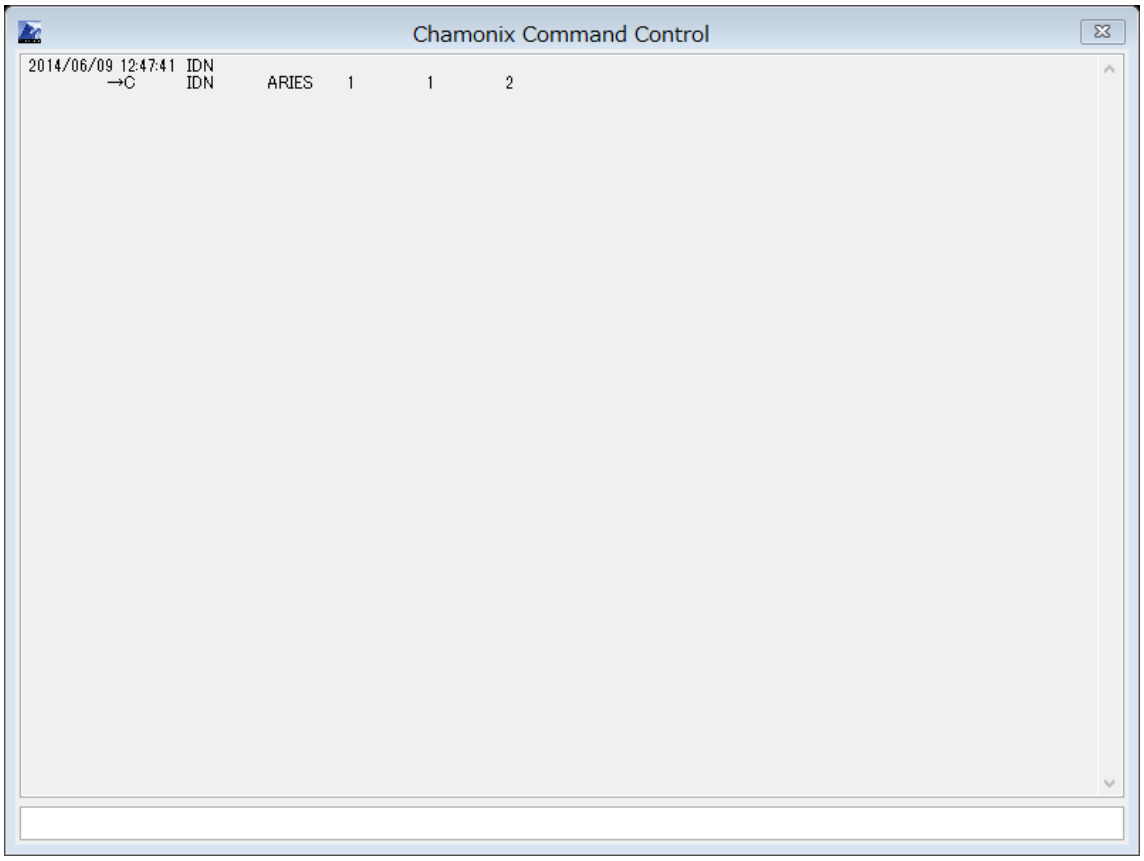
14.1. コマンドコントロールの起動



このボタンをクリックすると Chamonix Command Control が起動します。

Chamonix Command Control はコントローラへ送信するコマンドをキーボードから入力できます。

14.2. コマンドコントロールからコマンドを送信



送信ボックスにコマンドを入力し、「Enter」を押すとコントローラへコマンドが送信されます。RS-232C による通信に必要なヘッダーの STX 文字[20]とデリミタの CRLF 文字[21][22]は自動的に付加されます。リモートコントロールの書式については各々のコントローラの取扱説明書をご覧ください。入力は小文字で入力された場合も大文字に変換されます。

このウィンドウを使用してコマンドを送信した時間とコントローラからの返答がログウィンドウに表示されます。

アプリ側では一切のエラーチェックを掛けていません。正しいコマンドを入力してください。誤った送信文字列がコントローラに送られた場合、コントローラの仕様に従ってエラーが返されます。詳しくはコントローラの取

[20] 「STX」は「S」、「T」、「X」という３文字ではなく、「テキスト開始」を意味する１文字の制御文字です。

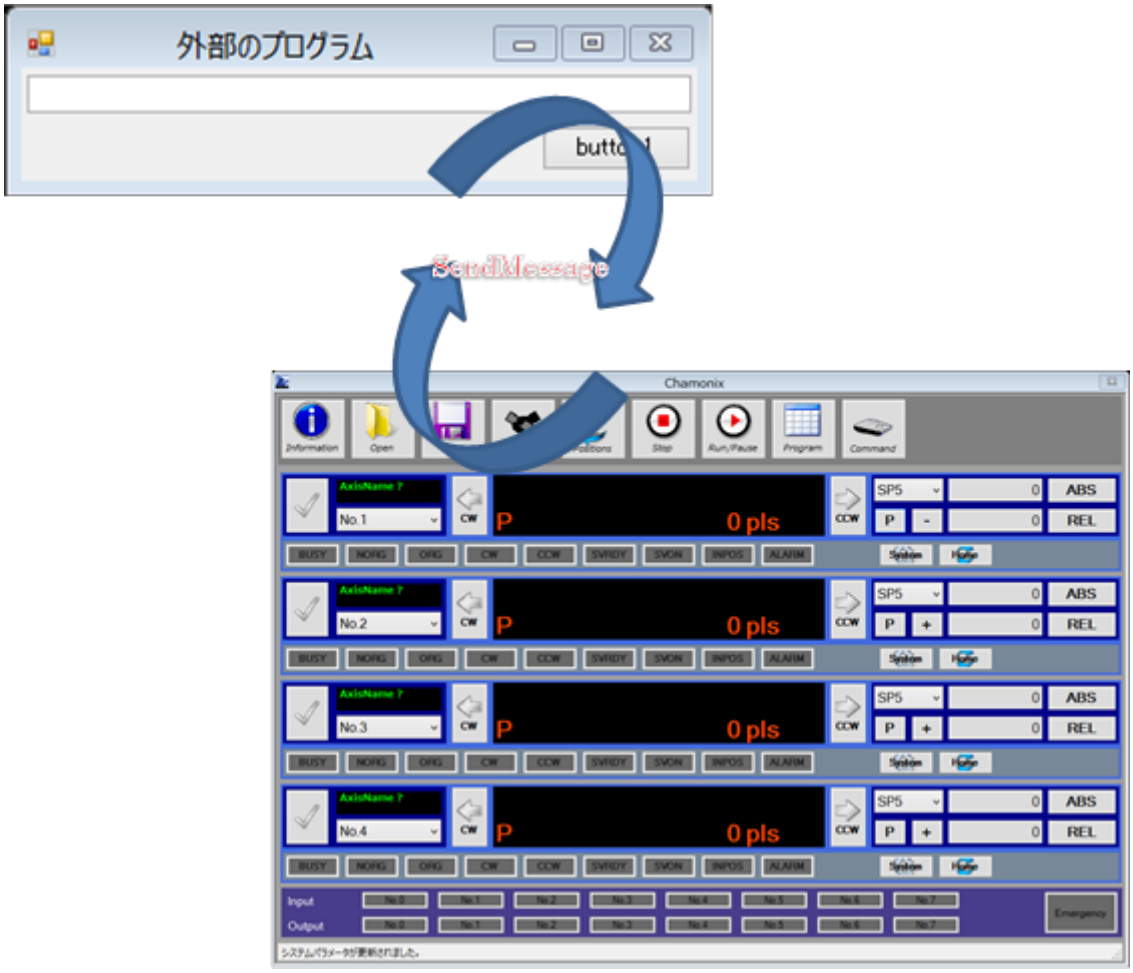
[21] 「CR」はローマ字二文字ではなく復帰を表す制御文字です。

[22] 「LF」は同様に改行を表す制御文字です。

説をご覧ください。

15. API を利用して外部から制御しましょう

15.1. WindowsAPI による外部制御



Chamonix は Windows 上の他のアプリから Windows API の WM_COPYDATA を利用して駆動の制御を行うことが可能です。

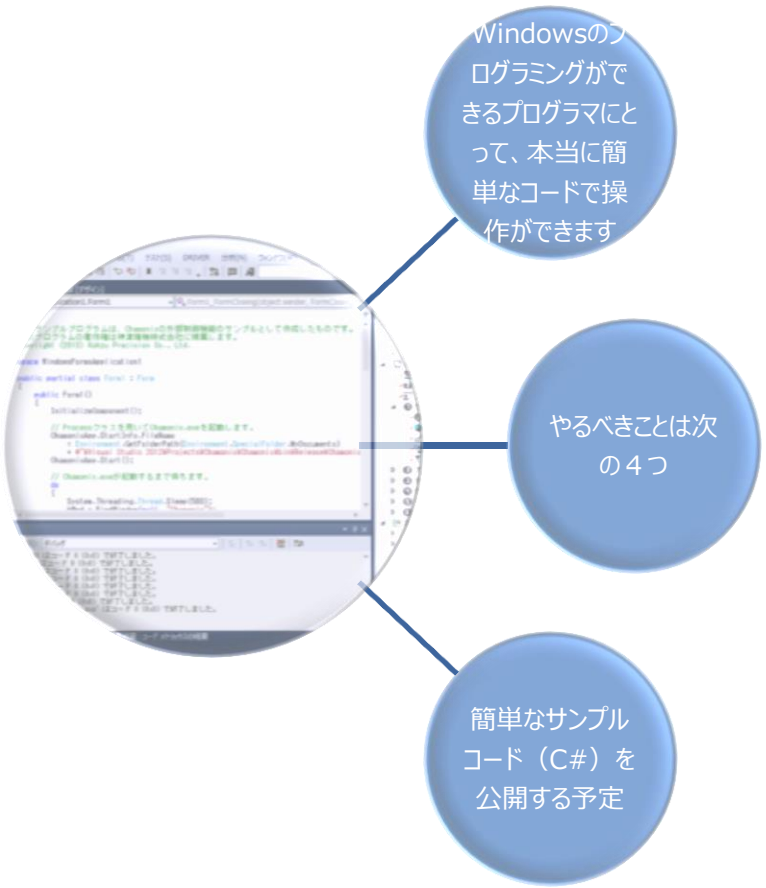
Windows API については Microsoft のウェブサイトなどを参照してください。

APS、RPS、ORG、STP のリモートコマンドに対応しています。

書式は ARIES のリモートコマンドの書式に準じます。

ただし返答方式のパラメータは省略されます。

15.2. 外部のアプリケーションからの制御



- Chamonixのウィンドウハンドラの取得
- USER32.DLLのSendMessage()のインポート
- COPYDATASTRUCT構造体に送信メッセージを入れる
- Chamonixへ送信する

外部のアプリではSendMessage()のコードを１行追加するだけで簡単にChamonixを制御できます。

15.3. 外部のアプリから APS コマンドを送る

```
IntPtr hWnd;

private int WM_COPYDATA = 0x004A;

USER32.dllのインポート

private void button1_Click(object sender, EventArgs e)
{
    // ChamonixのWindowハンドラを取得します。
    hWnd = FindWindow(null, "Chamonix");

    COPYDATASTRUCT copyStruct = new COPYDATASTRUCT();

    // 構造体stに値を入れます。
    copyStruct.dwData = (IntPtr)0;
    copyStruct.cbData = (uint)textBox1.Text.Length + 1;
    copyStruct.lpData = this.textBox1.Text;

    // SendMessageでChamonixにコマンドを送信します。
    SendMessage((IntPtr)hWnd, WM_COPYDATA, this.Handle, ref copyStruct);

    //Marshal.FreeCoTaskMem(lp);
    textBox1.Text = null;
}
/// <summary>
/// SendMessageで送信する構造体です。
/// </summary>
private struct COPYDATASTRUCT
{
    public IntPtr dwData;
    public uint cbData;
    public string lpData;
}
```

外部のアプリケーションから Windows API の SendMessage()関数[23]を使って、直接 Chamonix のウィンドウプロシージャにコマンドを渡します。

[23] 詳しくは Microsoft 社の MSDN の Web サイトをご覧ください。弊社への SendMessage 関数についてのお問い合わせはご遠慮ください。

SendMessage()関数は次のような構文を持っています。

SendMessage(HWND hWnd, UINT Msg, WPARAM wParam, LPARAM lParam)	
hWnd	Chamonix(メイン画面)のハンドルを入力
Msg	WM_COPYDATA（0x004A）
wParam	Chamonix からのメッセージを渡すウィンドウハンドル
lParam	送信するコマンド（COPYDATASTRUCT[24]） lParam.dwData：0 lParam.cbData：送信文字列のサイズ（例)13 lParam.lpData：送信文字列（例)APS1/9/10000

Chamonix に PM1 軸を、絶対位置管理で 10000pls の位置へ、速度テーブル No.9 で位置決めをさせたいとすると、
cbData に 12、
lpParam に「APS1/9/10000」を入力します。

返答方式のパラメータは省略されます。

[24]詳しくは Microsoft 社の MSDN の Web サイトをご覧ください。弊社への COPYSTRUCT 構造体についてのお問い合わせはご遠慮ください。

15.4. 外部のアプリから RPS コマンドを送る

```
IntPtr hWnd;

private int WM_COPYDATA = 0x004A;

USER32.dllのインポート

private void button1_Click(object sender, EventArgs e)
{
    // ChamonixのWindowハンドラを取得します。
    hWnd = FindWindow(null, "Chamonix");

    COPYDATASTRUCT copyStruct = new COPYDATASTRUCT();

    // 構造体stに値を入れます。
    copyStruct.dwData = (IntPtr)0;
    copyStruct.cbData = (uint)textBox1.Text.Length + 1;
    copyStruct.lpData = this.textBox1.Text;

    // SendMessageでChamonixにコマンドを送信します。
    SendMessage((IntPtr)hWnd, WM_COPYDATA, this.Handle, ref copyStruct);

    //Marshal.FreeCoTaskMem(lp);
    textBox1.Text = null;
}
/// <summary>
/// SendMessageで送信する構造体です。
/// </summary>
private struct COPYDATASTRUCT
{
    public IntPtr dwData;
    public uint cbData;
    public string lpData;
}
```

外部のアプリケーションから Windows API の SendMessage()関数[25]を使って、直接 Chamonix のウィンドウプロシージャにコマンドを渡します。

[25] 詳しくは Microsoft 社の MSDN の Web サイトをご覧ください。弊社への SendMessage 関数についてのお問い合わせはご遠慮ください。

SendMessage()関数は次のような構文を持っています。

SendMessage(HWND hWnd, UINT Msg, WPARAM wParam, LPARAM lParam)	
hWnd	Chamonix(メイン画面)のハンドルを入力
Msg	WM_COPYDATA（0x004A）
wParam	Chamonix からのメッセージを渡すウィンドウハンドル
lParam	送信するコマンド（COPYDATASTRUCT[26]） lParam.dwData：0 lParam.cbData：送信文字列のサイズ（例)13 lParam.lpData：送信文字列 （例)RPS1/9/100000

Chamonix に PM1 軸を、相対位置管理で 10000pls の位置へ、速度テーブル No.9 で位置決めをさせたいとすると、
cbData に 12、
lpParam に「RPS1/9/10000」を入力します。

返答方式のパラメータは省略されます。

[26]詳しくは Microsoft 社の MSDN の Web サイトをご覧ください。弊社への COPYSTRUCT 構造体についてのお問い合わせはご遠慮ください。

15.5. 外部のアプリから STP コマンドを送る

```
IntPtr hWnd;

private int WM_COPYDATA = 0x004A;

USER32.dllのインポート

private void button1_Click(object sender, EventArgs e)
{
    // ChamonixのWindowハンドラを取得します。
    hWnd = FindWindow(null, "Chamonix");

    COPYDATASTRUCT copyStruct = new COPYDATASTRUCT();

    // 構造体stに値を入れます。
    copyStruct.dwData = (IntPtr)0;
    copyStruct.cbData = (uint)textBox1.Text.Length + 1;
    copyStruct.lpData = this.textBox1.Text;

    // SendMessageでChamonixにコマンドを送信します。
    SendMessage((IntPtr)hWnd, WM_COPYDATA, this.Handle, ref copyStruct);

    //Marshal.FreeCoTaskMem(lp);
    textBox1.Text = null;
}
/// <summary>
/// SendMessageで送信する構造体です。
/// </summary>
private struct COPYDATASTRUCT
{
    public IntPtr dwData;
    public uint cbData;
    public string lpData;
}
```

外部のアプリケーションから Windows API の SendMessage()関数[27]を使って、直接 Chamonix のウィンドウプロシージャにコマンドを渡します。

[27] 詳しくは Microsoft 社の MSDN の Web サイトをご覧ください。弊社への SendMessage 関数についてのお問い合わせはご遠慮ください。

SendMessage()関数は次のような構文を持っています。

SendMessage(HWND hWnd, UINT Msg, WPARAM wParam, LPARAM lParam)	
hWnd	Chamonix(メイン画面)のハンドルを入力
Msg	WM_COPYDATA（0x004A）
wParam	Chamonix からのメッセージを渡すウィンドウハンドル
lParam	送信するコマンド（COPYDATASTRUCT[28]） lParam.dwData：0 lParam.cbData：送信文字列のサイズ（例)7 lParam.lpData：送信文字列（例)STP1/1

Chamonix に PM1 軸を緊急停止させたいとすると、
cbData に 7、
lpParam に「STP1/1」を入力します。

[28]詳しくは Microsoft 社の MSDN の Web サイトをご覧ください。弊社への COPYSTRUCT 構造体についてのお問い合わせはご遠慮ください。

15.6. 外部のアプリから ORG コマンドを送る

```
IntPtr hWnd;

private int WM_COPYDATA = 0x004A;

USER32.dllのインポート

private void button1_Click(object sender, EventArgs e)
{
    // ChamonixのWindowハンドラを取得します。
    hWnd = FindWindow(null, "Chamonix");

    COPYDATASTRUCT copyStruct = new COPYDATASTRUCT();

    // 構造体stに値を入れます。
    copyStruct.dwData = (IntPtr)0;
    copyStruct.cbData = (uint)textBox1.Text.Length + 1;
    copyStruct.lpData = this.textBox1.Text;

    // SendMessageでChamonixにコマンドを送信します。
    SendMessage((IntPtr)hWnd, WM_COPYDATA, this.Handle, ref copyStruct);

    //Marshal.FreeCoTaskMem(lp);
    textBox1.Text = null;
}
/// <summary>
/// SendMessageで送信する構造体です。
/// </summary>
private struct COPYDATASTRUCT
{
    public IntPtr dwData;
    public uint cbData;
    public string lpData;
}
```

外部のアプリケーションから Windows API の SendMessage()関数[29]を使って、直接 Chamonix のウィンドウプロシージャにコマンドを渡します。

[29] 詳しくは Microsoft 社の MSDN の Web サイトをご覧ください。弊社への SendMessage 関数についてのお問い合わせはご遠慮ください。

SendMessage()関数は次のような構文を持っています。

SendMessage(HWND hWnd, UINT Msg, WPARAM wParam, LPARAM lParam)	
hWnd	Chamonix(メイン画面)のハンドルを入力
Msg	WM_COPYDATA（0x004A）
wParam	Chamonix からのメッセージを渡すウィンドウハンドル
lParam	送信するコマンド（COPYDATASTRUCT[30]） lParam.dwData：0 lParam.cbData：送信文字列のサイズ（例)7 lParam.lpData：送信文字列（例)ORG1/9

Chamonix に PM1 軸を速度テーブル No.9 で原点復帰させたいとすると
cbData に 7、
lpParam に「ORG1/9」を入力します。

[30]詳しくは Microsoft 社の MSDN の Web サイトをご覧ください。弊社への COPYSTRUCT 構造体についてのお問い合わせはご遠慮ください。

15.7. 外部のアプリからプログラマブルグリッドのスタンバイを解除する

```
IntPtr hWnd;

private int WM_COPYDATA = 0x004A;

USER32.dllのインポート

private void button1_Click(object sender, EventArgs e)
{
    // ChamonixのWindowハンドラを取得します。
    hWnd = FindWindow(null, "Chamonix");

    COPYDATASTRUCT copyStruct = new COPYDATASTRUCT();

    // 構造体stに値を入れます。
    copyStruct.dwData = (IntPtr)0;
    copyStruct.cbData = (uint)textBox1.Text.Length + 1;
    copyStruct.lpData = this.textBox1.Text;

    // SendMessageでChamonixにコマンドを送信します。
    SendMessage((IntPtr)hWnd, WM_COPYDATA, this.Handle, ref copyStruct);

    //Marshal.FreeCoTaskMem(lp);
    textBox1.Text = null;
}
/// <summary>
/// SendMessageで送信する構造体です。
/// </summary>
private struct COPYDATASTRUCT
{
    public IntPtr dwData;
    public uint cbData;
    public string lpData;
}
```

外部のアプリケーションから Windows API の SendMessage()関数[31]を使って、直接 Chamonix のウィンドウプロシージャにコマンドを渡します。

[31] 詳しくは Microsoft 社の MSDN の Web サイトをご覧ください。弊社への SendMessage 関数についてのお問い合わせはご遠慮ください。

SendMessage()関数は次のような構文を持っています。

SendMessage(HWND hWnd, UINT Msg, WPARAM wParam, LPARAM lParam)	
hWnd	Chamonix(メイン画面)のハンドルを入力
Msg	WM_COPYDATA（0x004A）
wParam	Chamonix からのメッセージを渡すウィンドウハンドル
lParam	送信するコマンド（COPYDATASTRUCT[32]） lParam.dwData：0 lParam.cbData：送信文字列のサイズ lParam.lpData：送信文字列

Chamonix のプログラマブルグリッドを実行中に、SBY コマンドでスタンバイ状態になっているとき Message=指定子の文字列と lParam の文字列が一致したコマンドを受信すると、スタンバイ状態を解除します。

[32]詳しくは Microsoft 社の MSDN の Web サイトをご覧ください。弊社への COPYSTRUCT 構造体についてのお問い合わせはご遠慮ください。

15.8. 外部制御のサンプルプログラム

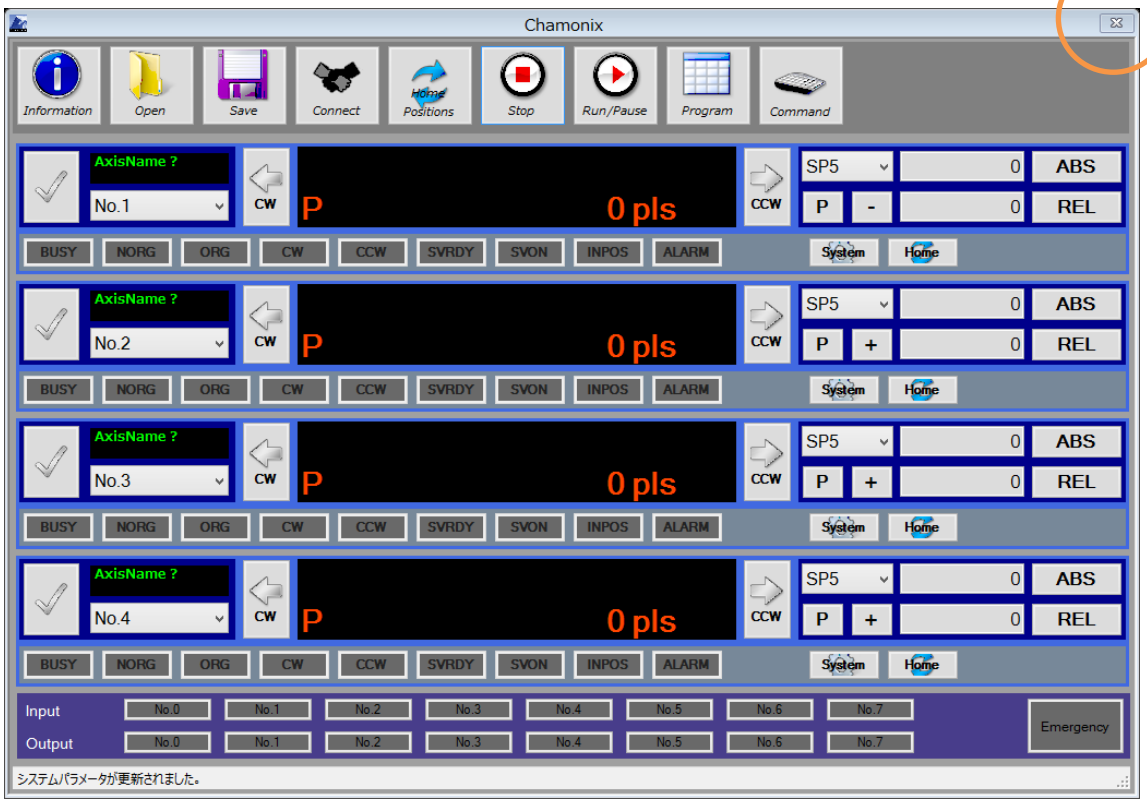
```
public partial class Form1 : Form
{
    public Form1()
    {
        InitializeComponent();

        // Processクラスを用いてChamonix.exeを起動します。
        ChamonixApp.StartInfo.FileName
            = Environment.GetFolderPath(Environment.SpecialFolder.CommonProgramFilesX86)
            + @"¥KOHZU¥Chamonix¥Chamonix.exe";
        ChamonixApp.Start();

        // Chamonix.exeが起動するまで待ちます。
        do
        {
            System.Threading.Thread.Sleep(500);
            hWnd = FindWindow(null, "Chamonix");
        }
        while (hWnd.ToInt32() == 0);

    }

    Process ChamonixApp = new Process();

    IntPtr hWnd;

    private int WM_COPYDATA = 0x004A;

    #region USER32.dllのインポート

    [DllImport("USER32.dll")]
    private static extern IntPtr SendMessage(IntPtr hWnd, int Msg, IntPtr wParam, ref COPYDATASTRUCT lParam);

    [DllImport("USER32.dll")]
    static extern IntPtr FindWindow(string lpClassName, string lpWindowName);

    #endregion

    private void button1_Click(object sender, EventArgs e)
    {
        // Chamonixのwindowハンドラを取得します。
        hWnd = FindWindow(null, "Chamonix");

        COPYDATASTRUCT copyStruct = new COPYDATASTRUCT();

        // 構造体stに値を入れます。
        copyStruct.dwData = (IntPtr)0;
        copyStruct.cbData = (uint)textBox1.Text.Length + 1;
        copyStruct.lpData = this.textBox1.Text;

        // SendMessageでChamonixにコマンドを送信します。
        SendMessage((IntPtr)hWnd, WM_COPYDATA, this.Handle, ref copyStruct);

        //Marshal.FreeCoTaskMem(lp);
        textBox1.Text = null;
    }
    /// <summary>
    /// SendMessageで送信する構造体です。
    /// </summary>
    private struct COPYDATASTRUCT
    {
        public IntPtr dwData;
        public uint cbData;
        public string lpData;
    }

    protected override void WndProc(ref Message m)
    {
        if (m.Msg.Equals(WM_COPYDATA))
        {
            try
            {
                COPYDATASTRUCT copyStruct = new COPYDATASTRUCT();
                copyStruct = (COPYDATASTRUCT)m.GetLParam(typeof(COPYDATASTRUCT));
                string text = copyStruct.lpData;
            }
            catch
            {
                throw;
            }
            textBox1.Text = copyStruct.lpData;
        }

        base.WndProc(ref m);
    }

    private void Form1_FormClosing(object sender, FormClosingEventArgs e)
    {
        ChamonixApp.CloseMainWindow();
    }
}
```

SendMessage を利用した外部のアプリケーションのサンプルコードは C#で Web から提供する予定です。サンプルコードはそれ自体を組み込んで実用に適するものとして設計しておりません。あくまでも SendMessage の利用法の一例として示しております。

サンプルコードをお客様が利用したことによるあらゆる責任を弊社としては負いかねます。 サンプルコードの著作権は神津精機株式会社に帰属します。

16. Chamonix を終了する

16.1. Chamonix を終了する



クローズボックスをクリックすると Chamonix は終了します。

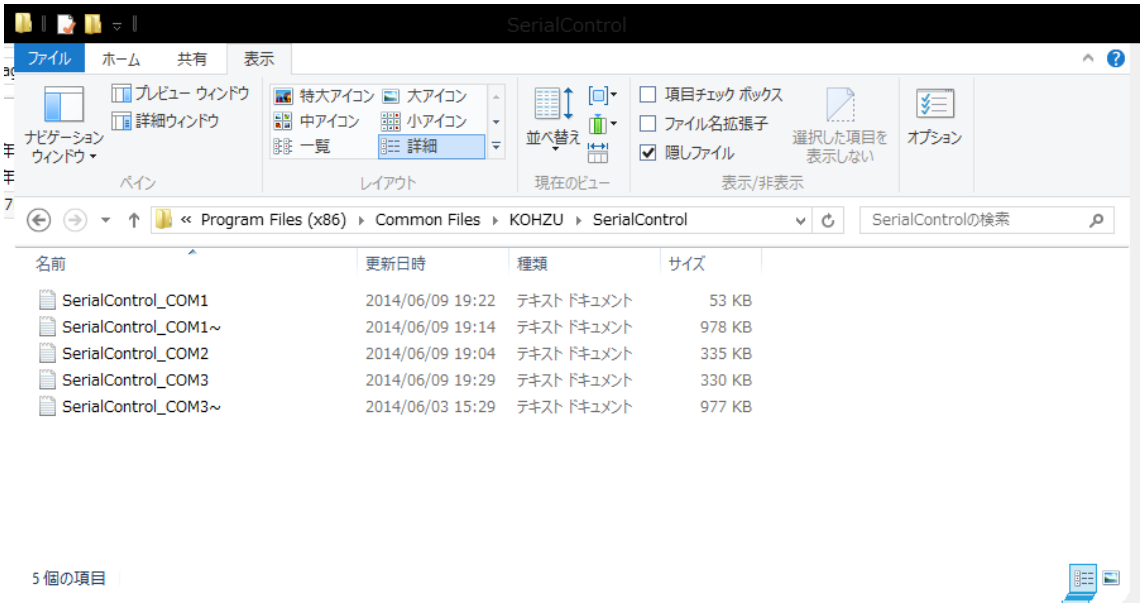
終了時に Common Files の Config.bin へ終了時のパラメータを書き込みます。COM ポート、ボーレート、各軸の換算係数、名称などが保存されます。コントローラのシステムパラメータは保存された値ではなくコントローラの設定値が優先されます。起動ファイルから起動されなかった場合、この Config.bin に保存された条件で起動します。

Chamonix が終了する際にコントローラの全軸に対して STP コマンドを送信します。

Chamonix で駆動をコントロールして、まだ駆動をしている最中に、Chamonix を落とした場合駆動はキャンセルされることを意味します。

17. その他

17.1. 通信ログの保存場所



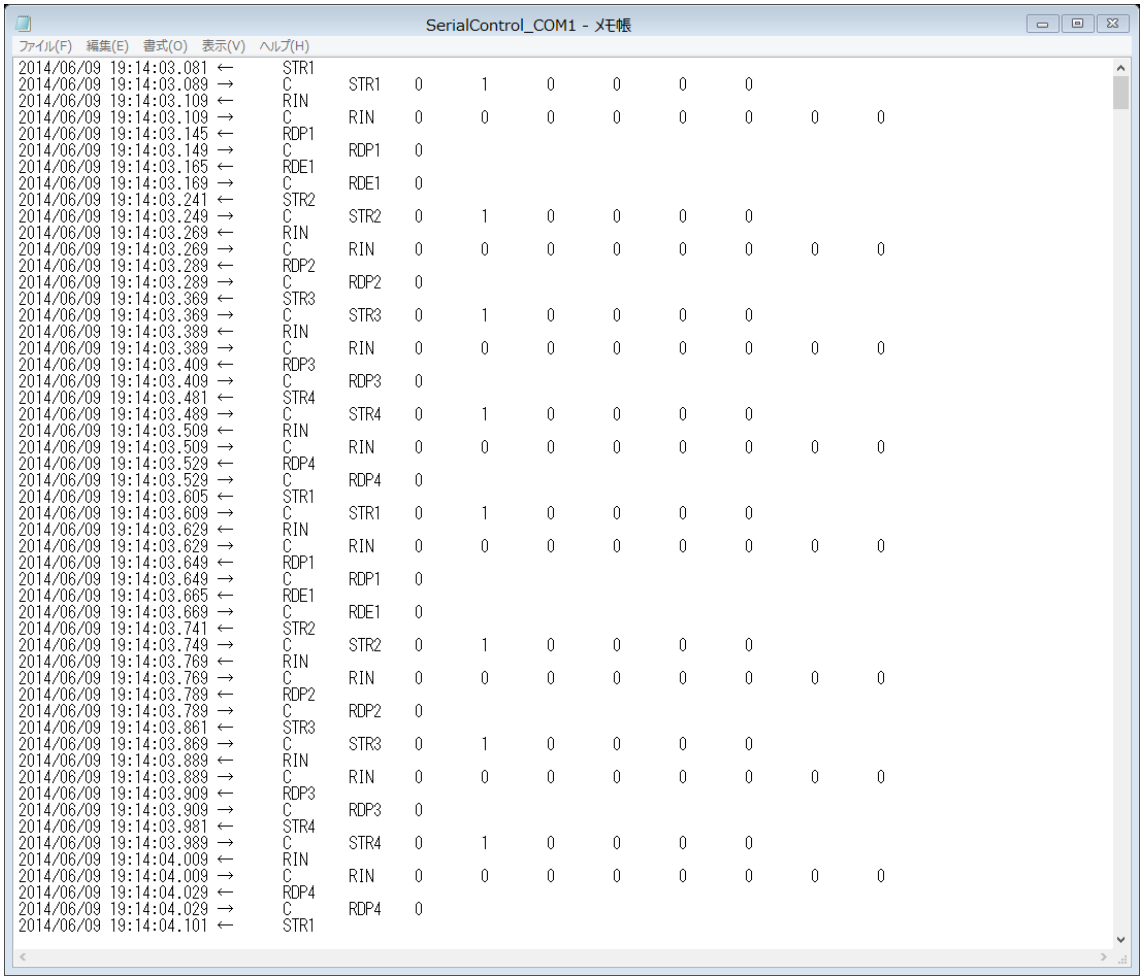
C:¥Program Files (x86)¥Common Files¥KOHZU¥SerialControl[33]フォルダ内にコントローラとの通信ログが残されています。

ログは 1MB を超える場合 1 回だけバックアップを取りリセットされます。

バックアップは次回のバックアップが行われる際に上書きされます。

[33] 32 ビット版の OS の場合 C:¥Program Files¥Common Files¥KOHZU¥SerialControl です。

17.2. 通信ログを確認

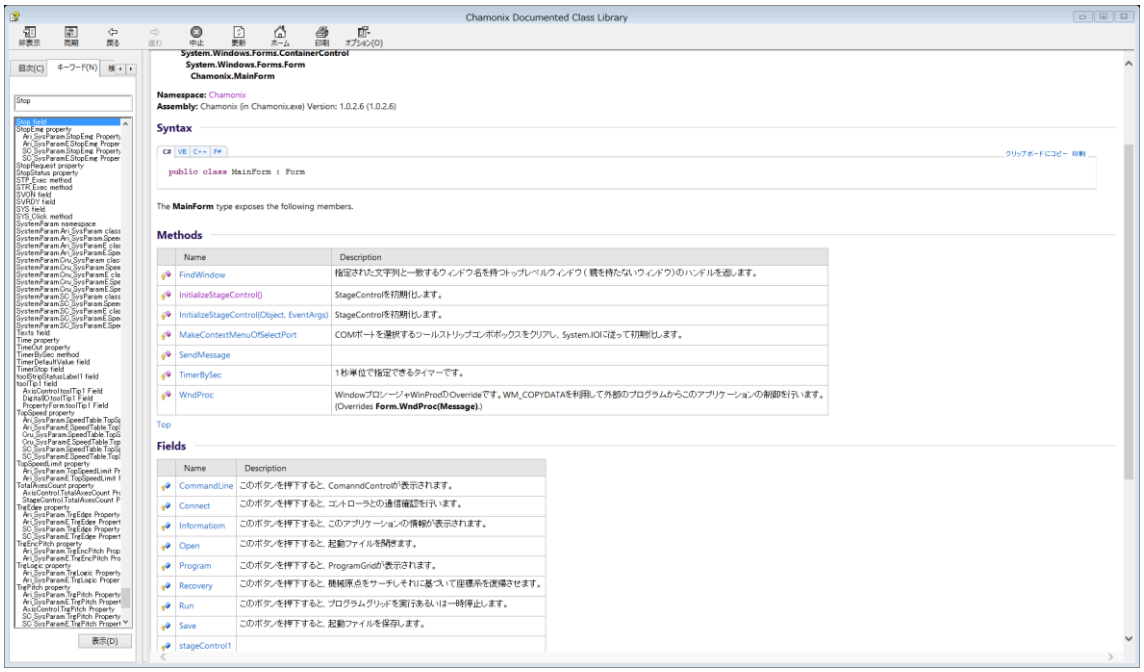


このログファイルにはコントローラとの最近の送受信が記録されています。

送受信の時間は、ミリ秒単位で記録されます。時間はシステムの時計を利用しているので正確ではありません。

問題が生じている場合、このファイルを解析することでより速いトラブルシューティングが期待されます。

17.3. 詳細なヘルプファイル



詳細な HTML ヘルプファイルが付属しています。

Help ファイルには Chamonix だけではなく、使用しているライブラリの各項目も記載されています。

Help ファイルは F1 キーを押下すると起動します。

索引

.cmxp, 32

1st Parameter, 66, 67, 68, 69, 70, 71, 72

2nd Parameters, 66, 67, 68, 71, 72

3rd Parameters, 66, 67, 68, 72

ABS, 48, 51, 67

API, 72, 76, 78, 80, 82, 84, 86

ARIES, 27, 28, 29, 35, 42, 52, 53, 59, 69, 72, 76

BUSY ランプ, 48, 50

CCW, 46

CCW 方向, 46

Chamonix, 8, 11, 13, 14, 18, 19, 21, 22, 23, 30, 33, 54, 56, 59, 63, 72, 74, 76, 77, 78, 79, 80, 81, 82, 83, 84, 85, 86, 87, 89, 92

COM ポート, 30, 31, 34, 35, 36, 89

Connect ボタン, 34, 35

CRUX, 23, 24, 25, 26, 35, 42, 54, 59

CSV, 59, 65

CW, 46

CW 方向, 46

DAT, 65, 76, 79, 81, 83, 85, 87

Emergency, 52

End, 62

Home, 39, 40

Home Positions, 39

Information ボタン, 37

LYNX, 27, 29, 35, 42

Microsoft .NET Framework, 11

Microsoft .NET Framework Full 4.0 以上, 11

Microsoft Windows 8.1, 14, 19, 20

Microsoft.NET4.0, 11

Open, 59

PAUSE, 63

PRINT, 60

Program Files(x86), 11

REL, 50, 51, 68

REL ボタン, 50

Repeat, 62

RS-232C, 23, 24, 27, 34, 75

RUN, 63

Setup, 12, 13

Setup.exe, 13

StageControl, 22, 35

Start, 62

STOP, 64

System, 22, 55, 57

Windows7®, 11

Windows8®, 11

WindowsXP®, 11

x64, 11

x86, 11, 19, 22, 33, 90

XML, 59, 65

アイコン, 30

アプリケーション, 8, 10, 22, 34, 58, 77, 78, 80, 82, 84, 86, 88

アンインストール, 20, 21

位置決め, 47, 48, 49, 50, 67, 68, 79, 81

印刷, 15, 60

インストール, 11, 12, 13, 14, 16, 17, 18, 19, 20, 21, 22, 30, 33

インストール CD, 12

エンコーダの現在位, 44

外部機器, 26, 29

仮想 COM ポート, 23

管理者権限, 12, 17, 23, 36

完了, 18

起動ファイル, 32, 43, 89

緊急停止, 64, 83

結線, 26, 29

現在位置表示, 44

原点復帰, 39, 40, 41, 66, 71, 85

神津精機, 88

コードサイニング証明書, 17

互換性, 55, 59

語長, 24, 27

コマンドコントロール, 74, 75

コントローラ, 8, 23, 34, 35, 36, 39, 45, 52, 55, 57, 58, 59, 62, 69, 71, 74, 75, 89, 90, 91

コントローラの取扱説明書, 34, 45, 75

コントロールパネル, 14, 20, 23, 36

コンボボックス, 42, 45

軸の名前, 43

システムパラメータ, 39, 45, 47, 49, 55, 57, 59, 66, 67, 68, 89

自動駆動, 62

使用許諾契約, 10, 13, 15, 16

使用許諾契約の確認, 13

使用許諾契約の確認と同意, 15

使用許諾契約の条項, 15

所属, 16

ステータス, 40, 71

ストップビット, 24, 27

制御画面, 34, 35

制御分解能, 47, 49

接続, 26, 29

接続確認, 34, 35

絶対位置, 47, 48, 50, 66, 67, 79

絶対位置管理, 47, 48, 50, 67, 79

前回終了時, 30, 33

全軸原点復帰, 39

相対位置, 49, 50, 68, 81

相対位置管理, 49, 50, 68, 81

速度テーブル, 45, 55, 66, 79, 81, 85

多軸駆動, 51

著作権, 88

通信設定, 24, 25, 27, 28, 30, 31

通信ログ, 90, 91

デバイスマネージャ, 23, 36

デフォルト, 30, 31, 33, 47, 49, 56, 58, 59, 65, 66, 67, 68

同意, 13, 15

入力, 15, 16, 17, 43, 47, 49, 57, 62, 66, 67, 68, 71, 72, 74, 75, 79, 81, 83, 85, 87

バージョン情報, 37

配線, 23

パラメータ設定, 55

パリティ, 24, 27

汎用出力, 69

非常停止, 52, 53, 54

プログラマブルグリッド, 61, 62, 66, 73, 86, 87

プログラムの追加と削除, 14

ヘルプファイル, 92

ボーレート, 30, 31, 32, 34, 35, 89

右クリック, 34

モータパルス, 44, 47, 49, 52

モータパルスの現在位置, 44

モダン UI, 31

ユーザーアカウントの制御, 17

ユーザー情報, 15, 16, 17

ユーザーズガイド, 8

ロータリースイッチ, 24, 25, 27, 28

対応 OS, 11

改訂履歴

年月日	版数	その他
2014 年 6 月 30 日	Rev. 1.0.0	
2014 年 10 月 30 日	Rev. 1.0.1	
2014 年 12 月 25 日	Rev. 1.0.2	

記録欄

ご購入日　　年　　月　　日

購入先

担当者　　電話番号

製造番号

特　記

・　・	
・　・	
・　・	
・　・	
・　・	
・　・	
・　・	
・　・	

KOHZU

Kohzu Precision Co., Ltd.

本社

〒215-8521

神奈川県川崎市麻生区栗木２－６－１５

Tel：**０４４－９８１－２１３１**

Fax：**０４４－９８１－２１８１**

E- mail： sale@kohzu.co.jp

Web Site： http://www.kohzu.co.jp/

大阪支店

〒532-0004

大阪市淀川区西宮原２－７－３８

新大阪西浦ビル２０２

Tel.：**０６－６３９８－６６１０**

Fax：**０６－６３９８－６６２０**